

VŠB - Technická univerzita Ostrava Fakulta  
elektrotechniky a informatiky  
Katedra informatiky a výpočetní techniky

Možnosti a použití CMS WordPress  
Possibilities of CMS WordPress

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student: **Václav Dobeš**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Možnosti a použití CMS WordPress  
Possibilities of CMS WordPress

Jazyk vypracování: čeština

### Zásady pro vypracování:

V dnešní době se při vytváření webových prezentací velmi často využívají různé redakční a publikační systémy. Mezi takovéto nejznámější systémy patří open source systém WordPress, který je určen pro vytváření webových stránek a blogů. Cílem této práce je popsat novinky z oblasti web designu se zaměřením na redakční systém WordPress, dále pak popsat postup při návrhu a použití moderních šablon pro WordPress, a to jak samostatně, tak s využitím dalších frameworků jako je například Bootstrap apod. V práci se také zaměříte na vytváření rozšiřujících pluginů, kde na vybraných příkladech demonstujete jejich základní možnosti a použití.

1. Nastudujte možnosti využití moderních technologií v prostředí redakčního systému Wordpress.
2. Popište postup tvorby nových šablon pro tento redakční systém s využitím aktuálních moderních technologií.
3. Zaměřte se také na možnosti responzivního designu pro následné využití jak pro desktopové, tak i mobilní verze.
4. Popište na ukázkových příkladech návrh a vývoj doplňujících pluginů (např. možnost vkládání a obarvování zdrojového kódu do stránek).
5. Výslednou práci upravte tak, aby sloužila i jako návod pro vytváření šablon a pluginů.

### Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Vedoucí bakalářské práce: **Ing. Martin Němec, Ph.D.**

Datum odevzdání: 28.04.2017

Guinda Gu

Am

prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.“

28.4.2017  
Datum

Další  
Podpis



## **Abstrakt**

Cílem práce je popsat novinky z oblasti web designu, především tedy responzivní návrh webové stránky, se zaměřením na redakční systém WordPress. Dále pak popsat postup při návrhu moderních šablon pro WordPress, a to jak samostatně, tak s využitím dalších frameworků s detailnějším zaměřením na framework Bootstrap. Šablona podporuje novou verzi jazyka HTML, tedy HTML5 a vlastnosti CSS3, a je na ní nastíněn postup tvorby webové stránky, který může sloužit jako návod pro komplexní tvorbu moderních webových stránek na bázi WordPressu a Bootstrapu s použitím nejnovějších webových technologií.

## **Klíčová slova**

Redakční systém, Wordpress, Bootstrap, HTML5, CSS3, responzivní web, tvorba webové stránky

## **Abstract**

The objective of the work is to describe web design news, especially the responsive web page draft, focusing on the WordPress content management system. In the next part will be description of the process of designing modern WordPress templates, both independently and using other frameworks with a more detailed focus on the Bootstrap framework. The template supports a new version of HTML, HTML5, CSS3 properties and it outlines a website creation process that can serve as a guideline for the complex creation of modern WordPress and Bootstrap websites using the latest web technologies.

## **Keywords**

Content management system, Wordpress, Bootstrap, HTML5, CSS3, responsive website, website creation

# Obsah

Seznam obrázků	8
Seznam tabulek	9
Seznam výpisů	10
Úvod	11
<b>1 Typy layoutů</b>	<b>12</b>
1.1 Fixní layout . . . . .	12
1.2 Fluidní layout . . . . .	13
1.3 Responzivní layout . . . . .	13
<b>2 Dnešní moderní web</b>	<b>15</b>
2.1 Přístupu k tvorbě . . . . .	15
2.1.1 Desktop first . . . . .	15
2.1.2 Mobile first . . . . .	16
<b>3 Fixní layout nebo responzivní layout</b>	<b>17</b>
3.1 Statistika . . . . .	17
3.1.1 Podíl mobilních zařízení na trhu . . . . .	17
3.1.2 Rozlišení obrazovek na internetu . . . . .	17
3.1.3 Zastoupení prohlížečů na internetu . . . . .	18
3.2 Srovnání na různých typech zařízení . . . . .	19
<b>4 Použité technologie</b>	<b>24</b>
4.1 HTML 5 . . . . .	24
4.2 CSS3 . . . . .	25
4.3 LESS . . . . .	25
4.4 Bootstrap . . . . .	26
4.4.1 Co je to Bootstrap . . . . .	26
4.4.2 Proč právě Bootstrap . . . . .	26
4.4.3 Implementace . . . . .	28
4.4.4 Dokumentace . . . . .	29
4.4.5 Customizace . . . . .	29
4.5 WordPress . . . . .	29
4.5.1 Co je to WordPress . . . . .	29
4.5.2 Proč zrovna WordPress . . . . .	29
4.5.3 Architektura WordPressu . . . . .	33
4.5.4 Databázový model WordPressu . . . . .	34
<b>5 Bootstrap v praxi</b>	<b>36</b>
5.1 Implementace Bootstrapu . . . . .	36
5.2 Hlavní pilíř Bootstrapu - Grid systém . . . . .	37
5.3 Připravené styly . . . . .	39

5.3.1	Typografie . . . . .	39
5.3.2	Tabulky . . . . .	40
5.3.3	Formuláře . . . . .	41
5.3.4	Tlačítka . . . . .	42
5.4	Komponenty . . . . .	43
5.4.1	Glyphicons . . . . .	43
5.4.2	Rozbalovací nabídka . . . . .	43
5.4.3	Skupina tlačítek . . . . .	44
5.4.4	Navigační lišta . . . . .	44
5.4.5	Progress bar . . . . .	44
5.5	Přízpůsobení Bootstrapu . . . . .	44
5.6	Tvorba webu pomocí Bootstrapu . . . . .	44
<b>6</b>	<b>WordPress v praxi</b>	<b>47</b>
6.1	Instalace WordPressu . . . . .	47
6.2	WordPress šablony . . . . .	48
6.2.1	Anatomie šablon . . . . .	48
6.2.2	Šablona stylů . . . . .	48
6.2.3	Soubor s funkcemi šablony . . . . .	49
6.2.4	Soubory šablon . . . . .	49
6.3	Tvorba šablony v praxi . . . . .	52
6.3.1	Šablony webu . . . . .	52
6.3.2	Soubor funkcí . . . . .	55
6.3.3	Šablony stránky . . . . .	57
6.4	Tvorba pluginu . . . . .	58
6.4.1	Co je to WordPress plugin . . . . .	58
6.4.2	Hlavička pluginu . . . . .	58
6.4.3	WordPress Hooks . . . . .	59
6.4.4	WordPress akční filtry . . . . .	60
6.4.5	Nastavení pluginu z prostředí administrace . . . . .	61
6.4.6	Komplexnější pluginy . . . . .	62
	<b>Závěr</b>	<b>63</b>
	<b>Literatura</b>	<b>64</b>
	<b>Příloha A - Podrobná architektura WordPressu</b>	<b>66</b>
	<b>Příloha B - Databázový model WordPressu</b>	<b>67</b>
	<b>Příloha C - Bootstrap komponenty</b>	<b>68</b>
	<b>Příloha D - Hierarchie WordPress šablon</b>	<b>72</b>
	<b>Příloha E - Výsledná podoba webového portálu</b>	<b>73</b>

## Seznam obrázků

1	Ukázka zmenšování fixního layoutu na úkor rozlišení daného zařízení . . . . .	12
2	Ukázka roztažení na rozlišení displeje u fluidního layoutu . . . . .	13
3	Ukázka přizpůsobení obsahu dle daného zařízení . . . . .	14
4	Media queries, přístup desktop first . . . . .	15
5	Media queries, přístup mobile first . . . . .	16
6	HTML podmíněný komentář . . . . .	16
7	Gray návštěvnosti dle měření NetMonitor . . . . .	17
8	Podíl různých typu rozlišení na internetu . . . . .	18
9	Podíl prohlížečů na internetu . . . . .	19
10	Ukázka fixního layoutu . . . . .	20
11	Ukázka responzivního layoutu . . . . .	20
12	Ukázka fixního layoutu na rozlišení 1024x768px . . . . .	21
13	Ukázka responzivního layoutu na rozlišení 1024x768px . . . . .	22
14	Ukázka layoutů na mobilním zařízení Samsung Galaxy S4 . . . . .	23
15	Použití frameworků v TOP jednom miliónu stránek. . . . .	27
16	Popularita systému WordPress a jeho největších konkurentů podle vyhledání v čase. . . . .	30
17	Procenta použití redakčního systému. . . . .	31
18	Použití redakčního systému v TOP milionu stránek. . . . .	32
19	Zjednodušená komunikace jednotlivých částí WordPressu. . . . .	33
20	Datový model systému WordPress. . . . .	34
21	Základní šablona s načtenými soubory . . . . .	37
22	Ukázka systému Grid . . . . .	39
23	Ukázka nastýlovaných nadpisů všech úrovní. . . . .	40
24	Ukázka použití nastýlovaných tabulek. . . . .	41
25	Ukázka použití nastýlovaných formulářů. . . . .	42
26	Ukázka použití nastýlovaných tlačítek. . . . .	43
27	Ukázka skládání šablony. . . . .	50
28	Ukázka vytvoření vlastního nastavení webu. . . . .	56
29	Ukázka přiřazení šablony ke stránce . . . . .	58
30	Demonstrace WordPress háčeků. . . . .	60
31	Demonstrace WordPress filtrů . . . . .	60
32	Výsledek použití pluginu.Vypnutý (vlevo), zapnutý (vpravo). . . . .	61
33	Nastavení pluginu z prostředí administrace . . . . .	62

## Seznam tabulek

1	Přepis statistiky z webu github. . . . .	27
2	Tabulky databázového modelu výchozí instalace. . . . .	35
3	Seznam tříd pro barevné odlišení buňek či řádků tabulky. . . . .	41

## Seznam výpisů

1	Vzor inforamcí o šabloně v souboru style.css. . . . .	48
2	Vykreslovací smyčka hlavní strany. . . . .	52
3	Registrace více navigací. . . . .	55
4	Zkracený výpis vytvoření vlastní uživatelské role. . . . .	57
5	Vzorová hlavička pluginu. . . . .	59

# Úvod

Digitální webová prezentace je v dnešní době nedílnou součástí každého subjektu. Proto se tato práce zaměřuje na rozebrání dnešních webových trendů a postupů při tvorbě moderní webové stránky. Moderní doba klade důraz na webové stránky přehledně dostupné skrze mobilní zařízení. Ukážeme si několik možných přístupů k tvorbě webové stránky, jejich klady a zápory, a také použitelnost v dnešní době. První kapitoly jasně ukáží, že doba statických webů je dávno pryč a podíl mobilních zařízení na webovém trhu prudce stoupá. Jsme tedy nuceni tvořit weby responzivní<sup>1</sup> ke spokojenosti velkého trhu mobilních návštěvníků. Práce se zaměří na tvorbu webové stránky pod záštitou redakčního systému WordPress a stylovacího frameworku<sup>2</sup> Bootstrap. Cílem práce bude popsat strukturu a rozebrat jednotlivé možnosti těchto nástrojů. Jelikož materiálů k tvorbě webových stránek na této platformě je v českém jazyce stále pramálo, práce nabídne i jednoduchý návod, jak s těmito nástroji pracovat a seznámí tak uživatele a začínající programátory s tvorbou a postupy při vytváření webových stránek na této platformě.

V další kapitole si tedy ukážeme nejnovější webové technologie, používané k tvorbě webových stránek. Zaměříme se tedy na nejnovější verzi HTML5<sup>3</sup>, jeho historii a použitelnost u moderního webu. Ukážeme a popíšeme si jeho velké výhody, nové funkce a možnosti. Podíváme se také na novou verzi stylovacího jazyku, konkrétně CSS3<sup>4</sup> a jeho obdobu v podobě koprocessoru LESS<sup>5</sup>. Ukážeme si jejich patřičné výhody a nové možnosti, které oproti svým starším verzím přinášejí.

Podrobněji si popíšeme framework Bootstrap a ukážeme jeho sílu a velké výhody oproti konkurenci. Vysvětlíme si, kde framework získat a hlavně jak jej začít používat. Dále se zaměříme na redakční systém WordPress a krátce také na jeho největší konkurenty. Podrobněji si popíšeme chování redakčního systému WordPress, jeho celkovou strukturu a architekturu.

V neposlední řadě se podíváme na framework Bootstrap a redakční systém WordPress z bližší perspektivy. Na vzorových příkladech si ukážeme použití těchto nástrojů a ukážeme si jejich funkcionalitu. Nastíníme jejich možnosti využití a zčásti nakoukneme na základní stavební kameny těchto nástrojů pro rychlou a snadnou tvorbu moderních webových stránek. Na příkladu portálu pro základní a mateřskou školu si ukážeme postup tvorby responzivní webové stránky na redakčním systému WordPress. Práce poskytne komplexní úvod do problematiky tvorby vlastních šablon pro redakční systém WordPress a poskytne tak čtenáři povědomí o struktuře a chování WordPress šablon.

---

<sup>1</sup>Způsob stylování HTML dokumentu, zaručující optimální zobrazení

<sup>2</sup>Softwarová struktura, slouží jako podpora při programování

<sup>3</sup>Verze značkovacího jazyka HTML sloužícího pro tvorbu webových stránek

<sup>4</sup>Třetí verze stylovacích kaskádových stylů CSS

<sup>5</sup>Knihovna, zavádějící do CSS vlastnosti známé z programovacích jazyků

# 1 Typy layoutů

Technologickému rozvoji mobilních zařízení se v posledních letech těší velký rozmach a jak je v oblasti informační technologie dobrým zvykem, docházelo k neustálému zmenšování těchto zařízení. Nyní máme proto na trhu nespočet kombinací rozlišení obrazovek zobrazovacího zařízení. Proto je všem těmto zařízením nutné předložit minimálně použitelné webové stránky. Bohužel, těchto webových aplikací, které se daným zařízením snaží nějakým způsobem přizpůsobit je stále málo. V této části si tedy představíme hlavní tři návrhové techniky pro tvorbu webu, které jsou nejčastěji k vidění na internetu.

## 1.1 Fixní layout

Jde o základní návrh, kdy je webová stránka postavena na jednom fixním rozlišení. Dříve se takto vytvářela většina webových stránek, nejčastěji na šířku okolo 960px, jelikož tohle rozlišení bylo u uživatelů nejrozšířenější. Pokud zobrazíme tento web na mobilu, tabletu, či jiném zařízení s malým rozlišením displeje, bude vypadat naprosto stejně jako na počítači, jenom zmenšený na velikost daného displeje. Takovéto weby bývají na těchto zařízeních těžce čitelné, takže je uživatel nucen zoomovat určité oblasti webu, na které se chce podívat blíže.

Aby se na mobilních zařízeních zobrazovaly stránky korektně a byly více uživatelsky přívětivé, nabízí některé weby svou mobilní verzi. Tato mobilní verze funguje většinou na doméně s prefixem “m”, tedy např. *www.m.domena.cz* a návštěvník webu je na ní automaticky přesměrován, pokud si webovou stránku zobrazuje právě na mobilním zařízení.

Mobilní verze webu je optimalizována na malé rozlišení telefonů a může obsahovat i jiný obsah než web klasicky. Mnohdy se dočkáme toho, že mobilní web je z důvodu větší optimalizace pro malé rozlišení výrazně osekán. Nevýhodou tohoto řešení je samozřejmě časová a finanční náročnost, jelikož jste přinucen vytvořit de facto dvě odlišné webové stránky, jednu pro klasické stolní desktopy a druhou pro mobilní zařízení. Z toho samozřejmě také vyplývá těžší správa webu, kdy jsme nuceni přidávat obsah zvlášť na fixní verzi webu a zvlášť pak na separátní mobilní verzi webu. Další nevýhoda spočívá v tom, že web již není přizpůsobený pro různá další rozlišení, jako třeba větší tablety či vysoká rozlišení u desktopů. Výhodou fixního layoutu může být rychlejší a obvykle levnější výroba a snadnější proces samotného návrhu webové stránky. Ukázka zdrojového kódu fixního layoutu je k nalezení jako Příloha 1 na CD/DVD.



Obrázek 1: Ukázka zmenšování fixního layoutu na úkor rozlišení daného zařízení

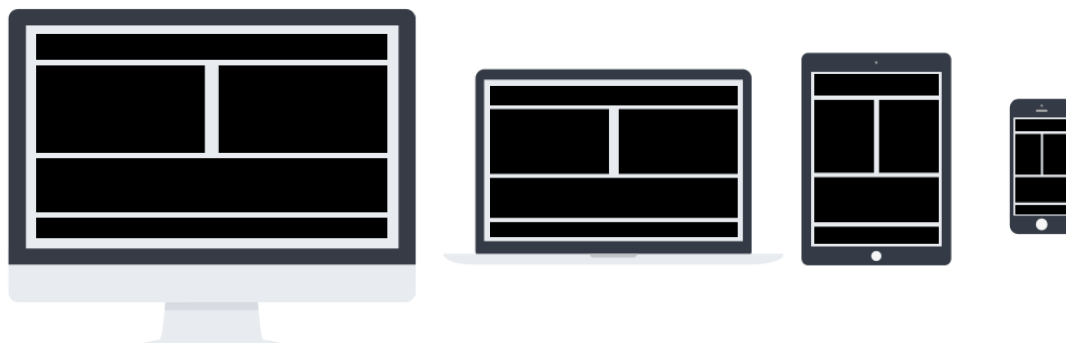


## 1.2 Fluidní layout

Méně používaná metoda tvorby webových stránek, která se dnes již moc často nevidí. Vznikla snahou o automatické přizpůsobení webu na dané rozlišení a tvoří tedy takový kompromis mezi fixním a responzivním layoutem.

Šířka webu se zde udává relativně, tedy v procentech. Při zobrazování webu na různých zařízeních se stránka vždy přizpůsobí rozlišení v daném poměru. Nicméně ani tohle řešení není ideální pro mobilní zařízení s malým rozlišením, jelikož dochází pouze ke zmenšení obsahu na dané rozlišení. Pokud tedy máme 3 sloupce textu na desktopu, budeme je mít i na mobilním zařízení, akorát výrazně zmenšené, tudíž se obsah opět stává nečitelným.

. Fluidní layout má tedy výhodu, že částečně webovou aplikaci přizpůsobí uživateli, avšak je zde náročnější testování obsahu v různých rozlišeních. Navíc u fluidního návrhu se stále nemění dispozice, tedy poloha jednotlivých prvků vůči sobě navzájem, takže na menších obrazovkách, jako mají mobily, se mohou některé prvky zmenšit tak, že jsou nepoužitelné. A v hodně případech může web vypadat zcela divně, což je způsobeno nepřizpůsobenými bitmapami vzhledem k ostatnímu obsahu stránky.



Obrázek 2: Ukázka roztažení na rozlišení displeje u fluidního layoutu

## 1.3 Responzivní layout

Responzivní layout je nejmodernější způsob designu webových aplikací využívající prvky CSS3. Layout vychází z myšlenky fluidního designu, ale přibývá zde možnost upravovat jednotlivé bloky pro různou škálu rozlišení. To znamená, že obsah se dynamicky mění na základě toho, z jakého zařízení uživatel na webovou aplikaci přistupuje. Ve zkratce, pokud uživatel přistoupí na webovou aplikaci z desktopu, můžeme mu nabídnout 3 sloupce obsahu, pokud z tabletu, nabídneme mu pouze 2 sloupce a pokud zobrazil aplikaci na mobilním telefonu, nabídneme mu pouze 1 sloupec obsahu, dynamicky přizpůsobený na velikost daného zařízení.

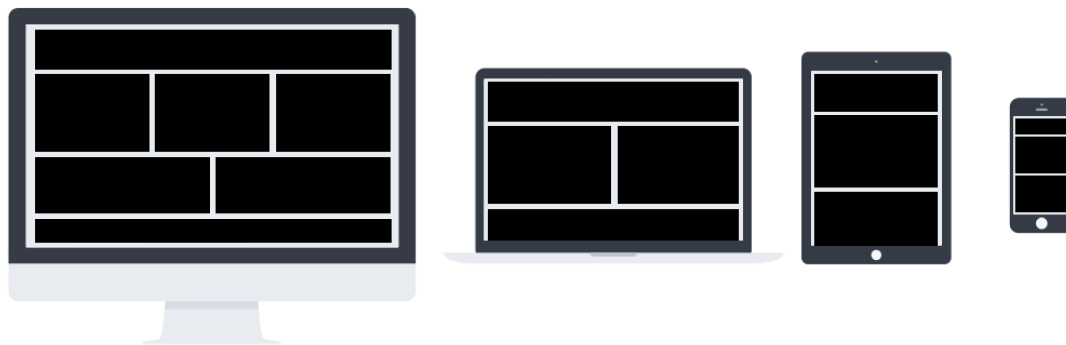
Responzivní design tedy využívá detekci rozlišení u klienta v prohlížeči, aby pro tohle rozlišení upravil rozvržení bloků na stránce. Také může pro různá rozlišení poskytovat různé obrázky, které jsou přizpůsobeny pro dané zařízení.

Mezi nevýhody patří především časová i finanční náročnost na vývoj a testování webové aplikace. Před samotným vývojem je nutné definovat rozložení ne na jednom, ale na několika zařízeních s různými rozlišeními displeje. V dalších fázích se jedná o náročnost při testování a optimalizaci webové

aplikace, kdy jsme často nuceni celý koncept dále upravovat, abychom docílili správné responzivní chování ve všech zařízeních. Mezi další nevýhodu patří rozdílné rozložení prvků na odlišných zařízeních. Toto může vést k tomu, že uživatelé, kteří byli zvyklí prohlížet si naši webovou stránku pouze na desktopu, mohou být při prvním setkání s responzivním webem na svém tabletu poněkud zmateni. Poté ovšem ocení výhody tohoto responzivního designu, které mírné zmatení prvních návštěvníku hravě vykompenzují.

K návrhu webu pomocí responzivní techniky již dnes existuje nespočet frameworků, které nám tvorbu značně usnadní a jedním z nich je i právě Bootstrap framework.

I největší světový vyhledávač Google <sup>6</sup> doporučuje z hlediska indexování výsledků vyhledávání responzivní webové stránky. Mezi důvody uvádí sjednocení URL adres pro všechna zařízení, které usnadňuje odkazování a sdílení stránek, a zkrácení času načítání stránky, jelikož není nutné zjišťovat typ zařízení a například mobilní zařízení přesměrovat na mobilní verzi webu. Uvádí také snížení zátěže jak pro provozovatele webových stránek, tak pro indexovacího robota. To je způsobeno tím, že indexovací robot není nucen procházet několik stránek se stejným obsahem optimalizovaných pro různá zařízení, ale jen jednu stránku s daným obsahem. Ukázka zdrojového kódu responzivního layoutu je k nalezení jako Příloha 2 na CD/DVD.



Obrázek 3: Ukázka přizpůsobení obsahu dle daného zařízení

---

<sup>6</sup><https://developers.google.com/webmasters/mobile-sites/mobile-seo/overview/select-config>

## 2 Dnešní moderní web

Jak jsme si již řekli, obrazovka mobilního zařízení je typicky menší než obrazovka monitoru, pro kterou je webová stránka navržena. Při prohlížení obsahu u fixního layoutu jsme nuceni stránku přiblížit, aby bylo písmo lépe čitelné. Po přiblížení zase ztrácíme nadhled na stránkou a hůře se na ní orientujeme. Stránku tedy oddálíme a takto pořád dokola. Webové inženýrství tento problém v zásadě řeší dvěma způsoby: verzí webové stránky pro mobilní zařízení nebo pomocí media queries<sup>7</sup>.

O mobilní webové stránce jsme si již pověděli v kapitole 1.1, nyní se proto zaměříme na vývoj webu pomocí media queries. Vývoj webu pomocí media queries (responzivní a adaptivní web) jde zcela jiným směrem než web navrženy speciálně pro mobilní zařízení. Cílem je vytvořit jeden web pro mobilní i desktop zařízení. Media queries je modul CSS3, který umožňuje podmíněné aplikování kaskádových stylů. Jeho zápis dovoluje omezovat aplikování CSS podle šířky či výšky obrazovky daného zařízení. Výsledkem je jedna webová responzivní stránka, která umí přizpůsobit svůj obsah danému zařízení. V případě, že potřebujeme webovou stránku často aktualizovat či přidávat nové věci, bude responzivní řešení vhodnější. Je to z důvodu jednodušší aktualizace webové stránky. A právě i díky tomu se dnešní moderní weby ubírají právě tímto směrem a proto se budu nadále věnovat pouze responzivní tvorbě webu.

### 2.1 Přístupu k tvorbě

Při tvorbě mobilního webdesignu lze zaujmout jeden ze dvou základních přístupů. Mobile first nebo desktop first. Každý má své výhody, ale také své nevýhody. Podstatou jejich odlišnosti je pořadí zařízení, pro která se design webové stránky vyvíjí. U mobile first jdeme tzv. cestou zdola, kdy první vytvořený design (CSS) webové stránky je optimalizován pro nejnižší (nejprimitivnější) zařízení s nejmenším displejem. Tím může být například osobní mobilní telefon, pro který je vytvořen čistě jedno-sloupcový layout. Poté se postupuje k sofistikovanějším zařízením jako je například smartphone, tablet, netbook nebo notebook až k osobním počítačům s monitory o vysokém rozlišení, přičemž původní jedno-sloupcový layout rozšiřujeme a obohacujeme o další grafické prvky a funkcionality. Naopak přístup desktop first upřednostňuje opačný směr vývoje, tedy od velkých monitorů osobních počítačů až po malé displeje mobilních telefonů.

#### 2.1.1 Desktop first

Programování s přístupem Desktop first tedy obnáší v první řadě vytvoření designu pro rozlišení osobních počítačů. Vytvořený CSS soubor se poté doplní o Media Queries podobně jako na ukázce obrázek 4.

```
/* Tady bude veškeré CSS pro desktop... */

@media screen and (max-width: 480px) {
  /* CSS pro malé displeje */
  @import 'mobile.css';
}
```

Obrázek 4: Media queries, přístup desktop first

<sup>7</sup><http://www.w3.org/TR/css3-mediaqueries/>

Dochází zde tedy k „přebíjení“ původního CSS kódu pro desktopová zařízení restrikcemi pro mobilní zařízení. Sám o sobě tento způsob programování není příliš „čistý“, a takový kód se navíc velmi těžko udržuje, protože při každé změně CSS pro desktopová zařízení se musí vytvářet případné restrikce směrem dolů. Dále také takto „dodáváme“ CSS kód v plném rozsahu i starším prohlížečům, které leckteré nové prvky (například CSS3) nepodporují a nepodporují například ani samotné media queries.

### 2.1.2 Mobile first

Naopak u mobile first se nejprve vytvoří základní CSS soubor pro zvolené zařízení na nejnižší úrovni, který je dále rozšiřován pro větší rozlišení.

```
/* Tady bude společná CSS základna pro křápy,  
mobilní zařízení a všechny ostatní:  
normalizace, typografie, barvy...*/  
  
@media screen and (min-width: 1024px) {  
  /* CSS pro velké displeje */  
}
```

Obrázek 5: Media queries, přístup mobile first

Z předcházející podkapitoly vyplývá, že základní CSS soubor platí i pro staré webové prohlížeče, které nepodporují media queries a tak svůj design nerozšíří o kód pro větší rozlišení. Z toho plyne výhoda i nevýhoda. Výhodou je možnost použití nejmodernějších prvků CSS3, které by na starých prohlížečích nefungovali, protože si je ani nenačtou, protože nepodporují media queries. Nevýhodou je naopak to, že existuje stále velké množství uživatelů, kteří používají webový prohlížeč Internet Explorer 8, který získá pouze základní vzhled z nejjednoduššího CSS, je proto potřeba zohlednit i tyto uživatele. Pro staré verze Internetu Exploreru lze snadno připravit speciální CSS a za pomoci podmíněných komentářů „vyladit“ vzhled stránek pro konkrétní verze, které jsou jinak velmi omezené co se jejich podpory CSS týče jako můžeme vidět na ukázce obrázku

```
<!--[if lte IE 8]>  
Zdrojový kód určený pouze pro Internet Explorer verze 8 a nižší.  
<![endif]-->
```

Obrázek 6: HTML podmíněný komentář

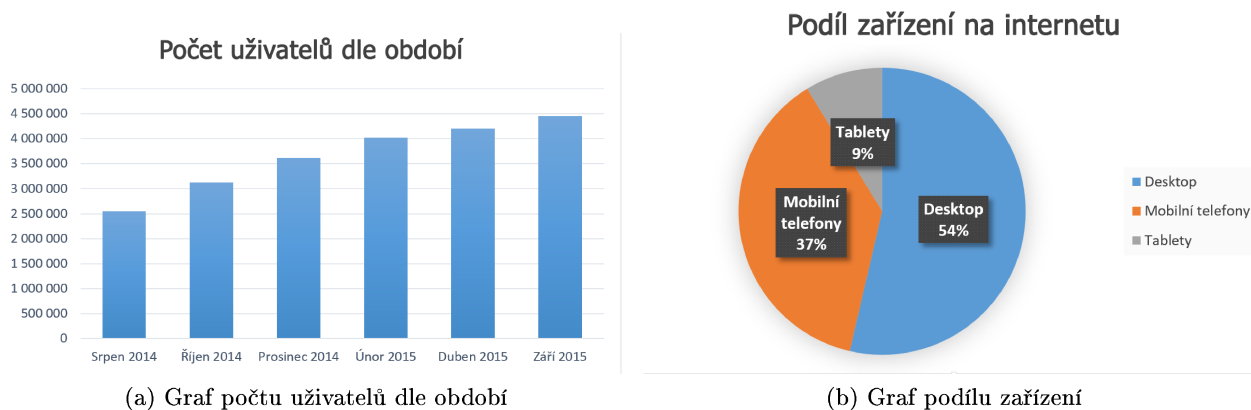
## 3 Fixní layout nebo responzivní layout

### 3.1 Statistika

#### 3.1.1 Podíl mobilních zařízení na trhu

Podíl uživatelů, kteří si chtějí prohlédnout webovou aplikaci přímo se svého mobilního zařízení stále roste. Zaměřil jsem se proto na studie české firmy NetMonitor, která sbírá údaje o návštěvnosti od největších webových projektů na českém internetu. NetMonitor poskytuje médiím v něm zapojeným a měřeným informace o návštěvnosti na úrovni reálných uživatelů, objektivní porovnání mezi všemi měřenými weby a rovněž informaci o demografickém profilu uživatelů těchto serverů a jejich sekcí. Upřednostnil jsem tuhle studii naproti celosvětově největší monitorovací studii od Google Analytics, jelikož NetMonitor studuje pouze ryze české webové stránky a tudíž poskytuje relevantnější informace právě pro Českou republiku.

Dle studií návštěvnost z mobilních zařízení začala rychle růst koncem loňského roku. Zatímco v srpnu 2014 dosahoval počet uživatelů navštěvujících měřené webové stránky z mobilních telefonů a tabletů 2,5 milionu, v říjnu jich bylo již přes 3 miliony. O tři měsíce později, začátkem roku 2015, překročil jejich počet rekordní hodnotu 4 milionů. V září tohoto roku se návštěvnost měřených webů z mobilních telefonů vyšplhal až na 4.5 miliónu reálných uživatelů viz graf 7a. Z tohoto trendu jasně vyplývá, že počet uživatelů, kteří si chtějí zobrazit webovou stránku na svém mobilním zařízení závratně roste a nejsme daleko od toho, kdy mobilní zařízení v počtu návštěvnosti zastíní klasické desktopy viz graf 7b. Proto si dále ukážeme, proč je právě responzivní web pro mobilní klienty přívětivější než web fixní.

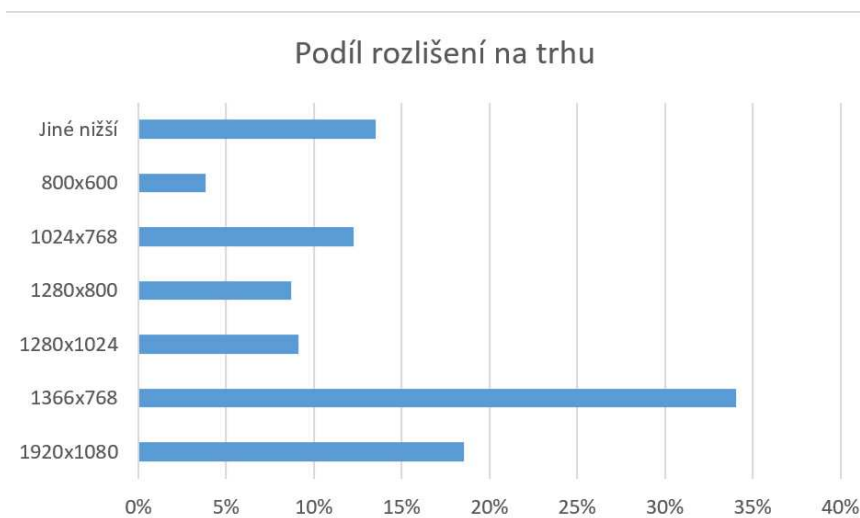


Obrázek 7: Graf návštěvnosti dle měření NetMonitor  
Zdroj dat: <http://www.netmonitor.cz/online-data-ola>

#### 3.1.2 Rozlišení obrazovek na internetu

Pro tuhle statistiku jsem si vybral 5 odlišných webů, které sbírají rozdílné statistiky na internetu. Jak je u těchto statistik zvykem, každá se od sebe nějak lišila. Tyhle získané údaje jsem zprůměroval a tím dostal výslednou statistiku podílu rozlišení zařízení, které se na internetu vyskytují. Škála těchto rozlišení je obrovská, proto jsem menší rozlišení jak 800x600 zahrnul do jedné skupiny, a tato skupina se stále ve statistice drží velmi obstojně. Jedná se především o levnější smartphony,

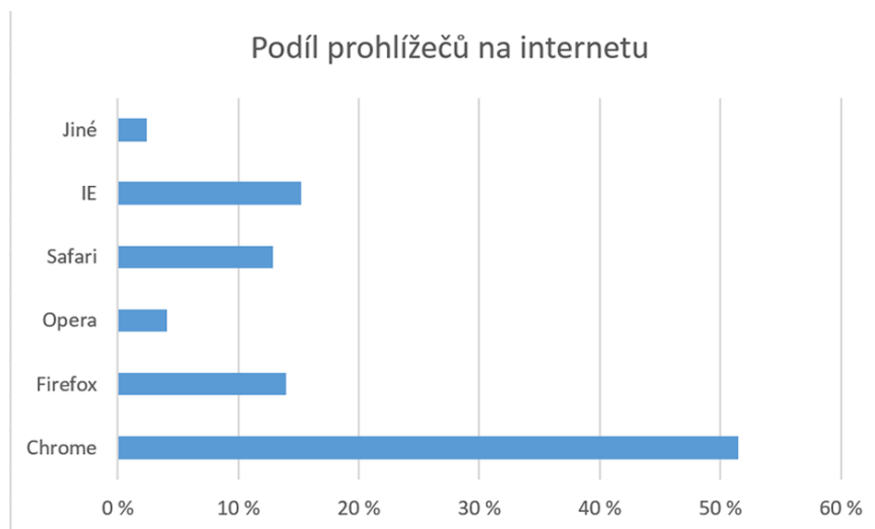
kteřé stále disponují rozlišením menším než právě nejmenší zmiňované 800x600 a jsou mezi běžnými uživateli vcelku rozšířené. Nejmenší podíl dosáhlo právě rozlišení 800x600, jelikož tohle rozlišení najdeme v dnešních smartphonech a tabletech jen velmi zřídka, tímto rozlišením se spíše myslí starší desktopy, kterých je dnes velmi málo. Naopak všechny statistické weby se shodly na jednom, a to, že nejrozšířenějším rozlišením na internetu je rozlišení 1366x768, což odpovídá menším notebookům, netbookům, tabletům a v dnešní době již také smartphonům. Větší rozlišení než udávané 1920x1080 jsem do statistiky nezahrnoval, jelikož se jednalo o pouhé desetiny procent a to především z důvodu, že desktopy s větším rozlišením než-li 1920x1080 stále nejsou mezi uživateli příliš rozšířené.



Obrázek 8: Podíl různých typu rozlišení na internetu

### 3.1.3 Zastoupení prohlížečů na internetu

Každý webový prohlížeč „chová“ více či méně odlišně, tedy vykresluje danou webovou stránku dle svých možností a nastavení. Právě proto nás, jako vývojáře zajímá také to, jaké prohlížeče jsou u uživatelů nejoblíbenější a tudíž na jaké prohlížeče se při optimalizaci stránek především zaměřit. Opět jsem si vyhlédl 5 různých webů s touto statistikou, které mi byly internetovou komunitou doporučeny jako věrohodné a tyto výsledky zprůměroval.

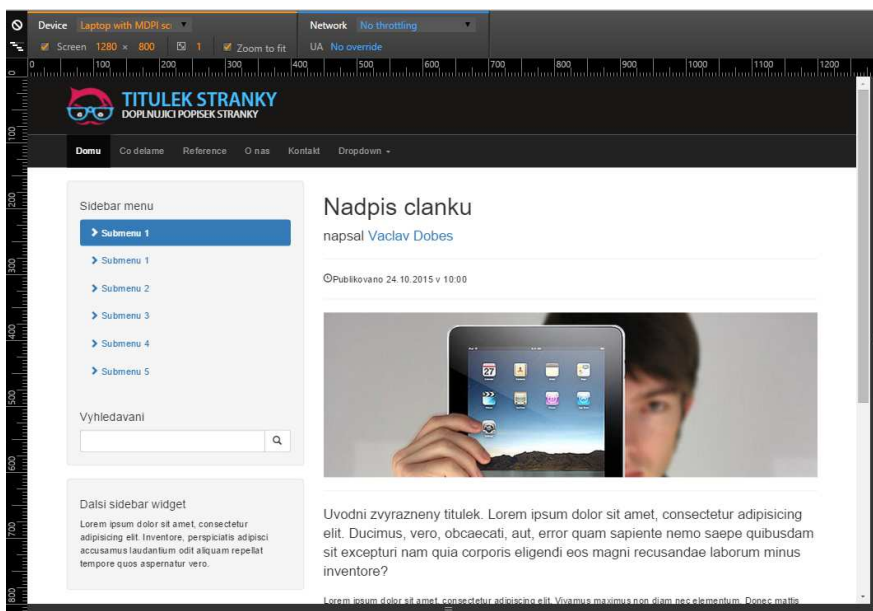


Obrázek 9: Podíl prohlížečů na internetu

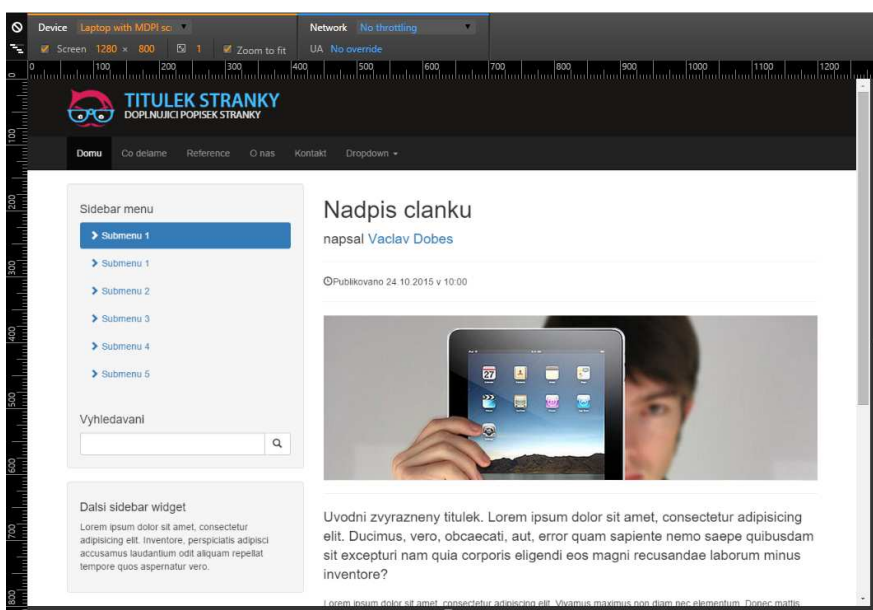
Z tohoto grafu jasně vyplývá, že Google Chrome je jedničkou mezi uživateli, a jeho podíl přesahuje 50% všech používaných prohlížečů na internetu. Naopak překvapením zůstává právě Internet Explorer, který se i přes svou špatnou pověst drží poměrně dobře ve srovnání s ostatními prohlížeči.

### 3.2 Srovnání na různých typech zařízení

Pokud si stránky zobrazíme na zařízení s rozlišením, pro které je fixní layout utvořen, nepoznáme mezi tímto layoutem a layoutem responzivním žádný výrazný rozdíl. Stránka je rozvržena přesně tak, jak bylo plánováno, vše je dobře viditelné a čitelné a pouhým okem nepoznáme, který layout je právě responzivní. Pro tohle porovnání nám skvěle poslouží integrovaný modul v Google Chrome a to Google Developer Tools, který umožňuje právě simulaci webové stránky napříč různými zařízeními. Nejprve si tedy naši vzorovou šablonu ukážeme na klasickém notebooku či stolním počítači s velmi rozšířeným rozlišením 1280x800px. Jelikož naše fixní šablona je tvořena na šířku 1140px, nebude proto problém zobrazit tento layout korektně viz obrázek 10. Na to si ověříme chování responzivního layoutu na totožném rozlišení. Jelikož responzivní layout nemá sebemenší problém přizpůsobit se danému rozlišení je výsledek téměř totožný viz obrázek 11.



Obrázek 10: Ukázka fixního layoutu

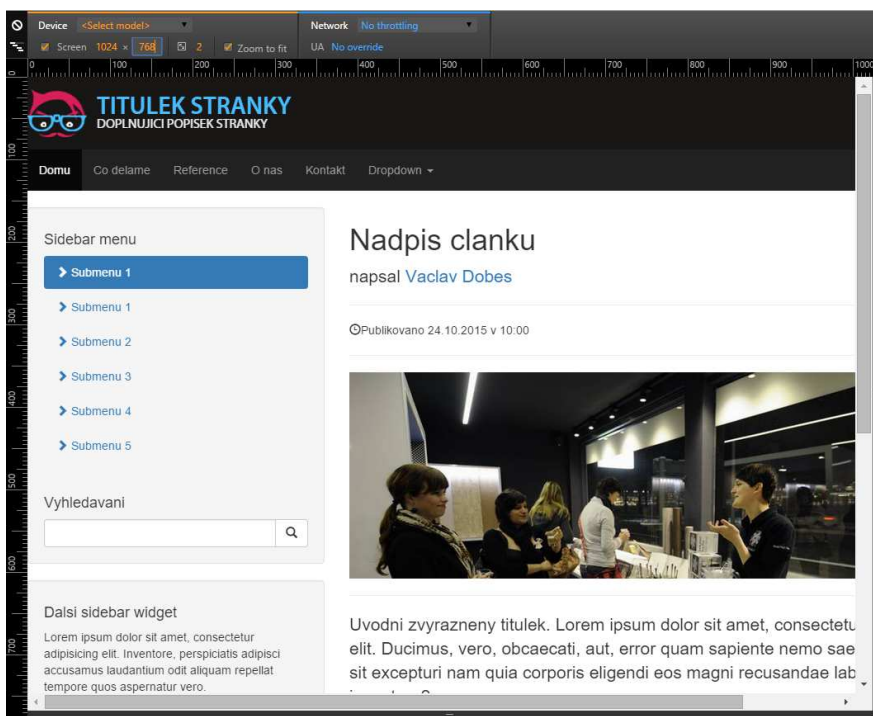


Obrázek 11: Ukázka responzivního layoutu

Problém u fixního layoutu ovšem okamžitě nastane ve chvíli, kdy se pokusíme daný layout zobrazit na nižším rozlišení. V našem případě je tedy nejnižší možné rozlišení, kdy se bude stránka chovat korektně, to s šířkou minimálně 1140px velkou. Pokud klesne šířka rozlišení pod 1140px, stránka se začne chovat nekorektně a různá zařízení budou vykreslovat rozvržení stránky různými způsoby. Například na stolním desktopu či notebooku s menším, ale stále vcelku používaným rozlišením 1024x768px, bude stránka oříznuta a zobrazí se nám vodorovný posuvník, naopak responzivní



layout se danému rozlišení hravě přizpůsobí a adaptuje svůj obsah na dané rozměry viz obrázky 12 a 13.

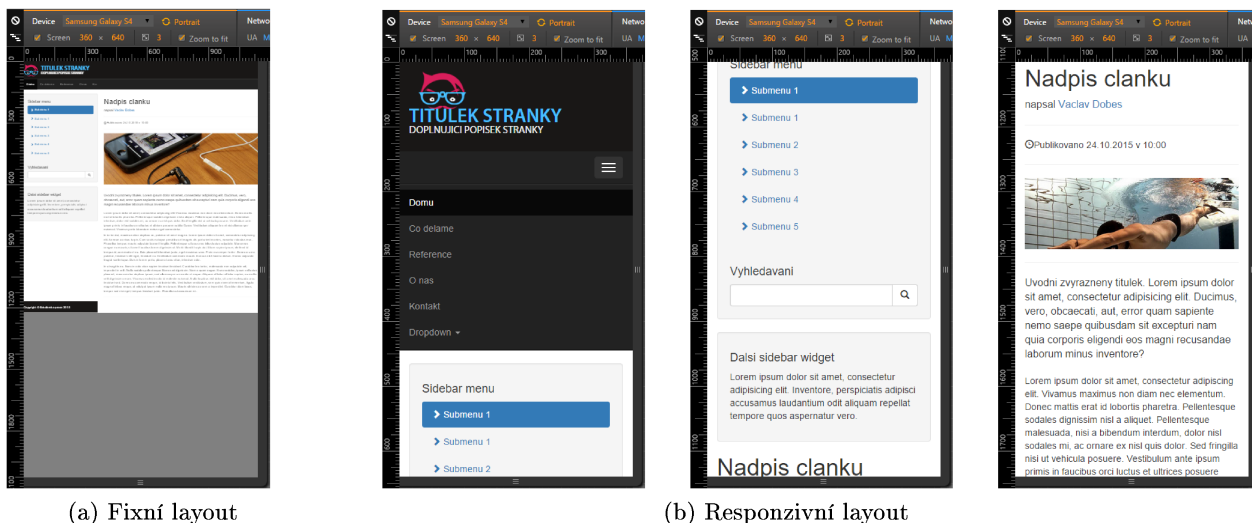


Obrázek 12: Ukázka fixního layoutu na rozlišení 1024x768px



Obrázek 13: Ukázka responzivního layoutu na rozlišení 1024x768px

Další problém nastane pokud budeme chtít zobrazit naši stránku na mobilním zařízení. Vycházíme samozřejmě z toho, že máme fixní layout bez separátní mobilní verze, o jejíž nevýhodách jsem se již zmínil v kapitole 1.1. Jako vzor jsem vybral rozlišení telefonu Samsung Galaxy S4 a to 360x640px, který nám Google Developers Tools nabízí. Fixní layout je na tomto rozlišení automaticky zmenšen a tudíž se stává absolutně nečitelným a uživatel je donucen si layout neustále zoomovat<sup>8</sup> viz obrázek 14a. Pro mobilního uživatele se proto taková webová stránka stává otravnou a z velkou pravděpodobností dá příště přednost konkurenční stránce, která je naopak mobilním zařízením přizpůsobená. Naopak pokud se podíváme na responzivní layout, můžeme si všimnout, že stránka se hravě adaptovala na dané rozlišení a je na mobilním. Můžeme si tedy povšimnout, že jednotlivé segmenty stránky se naskládaly pod sebe a roztáhly na šířku rozlišení, čímž se webová stránka stává krásně čitelnou a uživatelsky velmi přívětivou viz obrázek 14b.



Obrázek 14: Ukázka layoutů na mobilním zařízení Samsung Galaxy S4

<sup>8</sup>Přiblížení daného místa pomocí dotyku (obdoba CTRL + kolečko\_myši ze stolních PC)

## 4 Použité technologie

### 4.1 HMTL 5

HTML5 je nástupce HTML 4.01 a po ukončení vývoje XHTML 2 hlavní cesta pro tvorbu moderního webu. Specifikace je to obrovská a lidé často mluví o HTML5, ale myslí pouze jeho malý kousek. Ta totiž nepokrývá jen sémantiku a markup, ale zabývá se JavaScriptovými API, offline fungováním aplikací, kreslením v prohlížeči, a vůbec řeší mnoho věcí, které potřebujeme pro vytváření webových aplikací. [1]

HTML verze 5 se od verze 4 liší novými, zkrácenými a rychlejšími zápisy značek. Autoři dávají důraz na jednoduchost a zároveň účinnost. HTML5 je též možné vytvořit aplikaci, která funguje v prohlížeči i tehdy, když uživatel nemá internetové připojení, a která ukládá data do lokálního úložiště na uživatelské počítači. Je-li internetové připojení k dispozici, může aplikace synchronizovat data se vzdáleným serverem.

Má-li aplikace fungovat v offline módu, je nutné vytvořit tzv. cache manifest, což je seznam souborů, jež mají být přístupné lokálně. Jméno souboru s tímto seznamem se uvede jako vlastnost prvku HTML, např. `<html manifest="cache.manifest">`. Jako perzistentní úložiště lze využít buď relační databázi, nebo asociativní pole `localStorage` přístupné přes objekt `window`. Kromě `localStorage` existuje navíc objekt `sessionStorage`, v němž je možné uchovávat data po dobu trvání sezení.

HTML5 se také zaměřilo na sémantiku webových stránek a převážně na zvýšení přehlednosti zdrojového kódu. Většina stránek je dnes tvořena obvyklými částmi, jako je hlavička, různé sloupce a patička. Tyto části jsou odlišeny pomocí prvku `div` a jeho vlastností `id` nebo `class`, protože v současné verzi HTML 4 žádné speciální prvky na toto rozlišení nejsou. V HTML5 jsou proto zavedeny tyto nové značky, které jsou určeny na strukturování stránek:

- `<section>` – představuje části stránky, např. kapitoly
- `<article>` – představuje nezávislé části stránky, např. články nebo komentáře
- `<main>` – představuje hlavní obsah stránky
- `<aside>` – představuje části stránky, které jen nepatrně souvisí se zbytkem stránky, např. poznámky stranou V návrhu WHATWG:
- `<header>` – představuje hlavičku, může obsahovat například nadpis nebo navigační odkazy
- `<footer>` – představuje patičku, může obsahovat informace o autorovi nebo autorských právech
- `<nav>` – představuje část stránky, která je určena k navigaci
- `<figure>` – představuje samostatný obsah stránky, který doplňuje hlavní stat, ale není její součástí, např. obrázek, graf, video nebo ukázka kódu
- `<figcaption>` – představuje popis pro `<figure>`

Ve starších verzích (HTML v4.01 a starší) neexistují prostředky, které by úspěšně pracovaly s vkládáním multimédií, proto jsou využívány různé pluginy nebo Flash. Tento problém se HTML5 také snaží vyřešit a zavádí proto tyto značky:

- `<video>` – vkládání videa

- `<audio>` – vkládání audia
- `<source>` – definuje alternativní verze videa nebo audia souboru, z nichž si prohlížeč vybírá ty, které podporuje
- `<track>` – vkládá stopu pro `<video>` či `<audio>`

Další novinkou je pak prvek `<canvas>`, který umí interpretovat vektorovou grafikou s možností vkládat obrázky. Dá se na něj kreslit prostřednictvím skriptovacích jazyků na straně klienta s použitím kontextových metod, mj. pomocí grafických primitiv. Použití jako např. grafy reagující na ovládání uživatele.

## 4.2 CSS3

CSS, Cascading Style Sheet, je jazyk definování vizuálního stylu zobrazování HTML elementů na webu. V kombinaci s HTML dokáže poskytnout internetovému prohlížeči informaci o tom, jak má zobrazit jednotlivé elementy dokumentu a celý dokument jako celek. CSS dokáže aplikovat různé vlastnosti na elementy, jako například orámování, vnější a vnitřní odsazení, výšku, šířku, barvu, pozadí, typ písma a mnoho dalších. Nové CSS3 přináší novou koncepci modulární struktury, rozdělenou do více souborů (modulů), díky čemuž umožňuje definovat funkcionalitu odděleně a lépe jí tak spravovat. Těchto modulů má CSS3 zatím něco přes 40, patří mezi ně například: [2]

- selektory
- pozadí a orámování
- textové efekty
- 2D/3D transformace
- animace
- sloupcové rozdělení dokumentu

## 4.3 LESS

LESS je CSS koprocessor, což znamená že rozšiřuje kaskádové styly a přidává funkcionality jako definování proměnných, funkcí, jmenných prostorů a oblastí platnosti a mnoho jiných užitečných rozšíření. To znamená, že naše CSS soubory jsou jednodušší pro správu, rozšiřitelnosti a opakovatelnou použitelnost.

Běžné použití preprocesoru je následující. Řekněme, že používáme v kaskádových stylech určitý hexadecimální kód barvy, a tato barva je použita na více místech v souboru. Pokud se rozhodneme, že chceme tuto barvu změnit kompletně v celém souboru, musím najít všechny výskyty této barvy a nahradit jejich kódem nové barvy. S použitím LESS se barva uloží do proměnné a při potřebě změny barvy se jednoduše změní hodnota proměnné, zkompileje se LESS soubor a výsledkem kompilace je upravený CSS soubor s novou barvou.

LESS se oproti klasickému CSS nabízí mnoho vylepšení jako jsou:

- Proměnné - Vám umožní specifikovat často používané hodnoty na jednom místě a následně je jen využívat v stylesheetu. Změny těchto hodnot jsou tak záležitostí editace jednoho řádku.

- Mixiny - Umožňují vložit všechny vlastnosti třídy do jiné třídy, a to jednoduše specifikací jejího názvu uvnitř jiné třídy. Berte je jako proměnné, ovšem pro celé třídy. Mixiny se rovněž mohou chovat jako funkce a pracovat s argumenty, viz následující ukázka.
- Vnořená pravidla - Místo vytváření selektorů s dlouhými názvy ke specifikaci dědičnosti můžete v LESS použít vnoření selektorů do jiných. Díky tomu bude zápis přehlednější a stylesheet kratší.
- Funkce a výpočty - Jsou všechny elementy ve vašich stylech v určité proporci k jiným elementům? Pomocí výpočtů v LESS můžete přidávat, odečítat, dělit a násobit hodnoty a barvy. Získáváte tedy výkonný nástroj pro vytvoření komplexních vztahů mezi vlastnostmi jednotlivých selektorů. Funkce vám umožňují pracovat s JavaScriptem a manipulovat s hodnotami, jak potřebujete.

## 4.4 Bootstrap

### 4.4.1 Co je to Bootstrap

Jedná se moderní frontend framework, který slouží převážně k vývoji responzivních webových stránek. Tento framework původně vytvořili Jacob Thornton a Mark Otto v roce 2010, jako framework pro svůj systém Twitter. Později framework vypustili do světa a nabídli jej tak i ostatním vývojářům.

Bootstrap je open-source řešení pro tvorbu responzivního uživatelského rozhraní webových aplikací založených na jQuery knihovně. Jeho mobile-first přístup a jedna z nejaktivnějších komunit mu dodávají velký potenciál. Bezespornou výhodou je také strmá křivka učení, zapříčiněná kvalitní dokumentací a nízkou náročností v začátcích. Nabídka tohoto frameworku není jen o CSS stylech, disponuje i širokou škálou JavaScriptových doplňků a komponent.

Základ Bootstrapu tvoří kompilované a minimalizované soubory (`bootstrap.*`, `bootstrap.min.*`), které je možné použít ve kterémkoliv webovém projektu. Jedná se modulární nástroj složený převážně z LESS stylůpisů, kde všechny Bootstrap komponenty můžeme najít ve stylůpisu „`bootstrap.less`“. Vývojáři si tedy mohou Bootstrap knihovnu přizpůsobit k obrazu svému převážně výběrem komponent, které chtějí pro svůj projekt využít. Grafika je komprimovaná pomocí ImageOptim<sup>9</sup>.

### 4.4.2 Proč právě Bootstrap

Za největší konkurenci právě zmiňovaného Bootstrapu se považují dva responzivní CSS frameworky. První z nich je framework Foundation<sup>10</sup> od společnosti Zurb, která se zabývá designem a tvorbou webových stránek, kde byl tento framework také nejdříve používán jako interní pomůcka pro firemní produkci a teprve v roce 2011 vypuštěn v podobě open-source řešení. A druhý z nich, framework Skeleton<sup>11</sup>, který vytvořil Dave Gamache jako open-source řešení pro jednoduchou tvorbu responzivního designu, jehož součástí jsou pouze základní funkce a UI elementy. Bohužel, leč se posledně jmenovaný Skeleton řadí mezi největší konkurence, jeho oficiální vývoj byl ukončen začátkem roku 2015, proto je srovnávání s Bootstrapem zbytečné. Nicméně ačkoliv byl jeho vývoj ukončen, má tento framework stále velkou komunitu svých příznivců a tak je i nadále u vývojářů docela oblíben.

<sup>9</sup><https://imageoptim.com/mac>

<sup>10</sup><http://foundation.zurb.com>

<sup>11</sup><http://getskeleton.com>

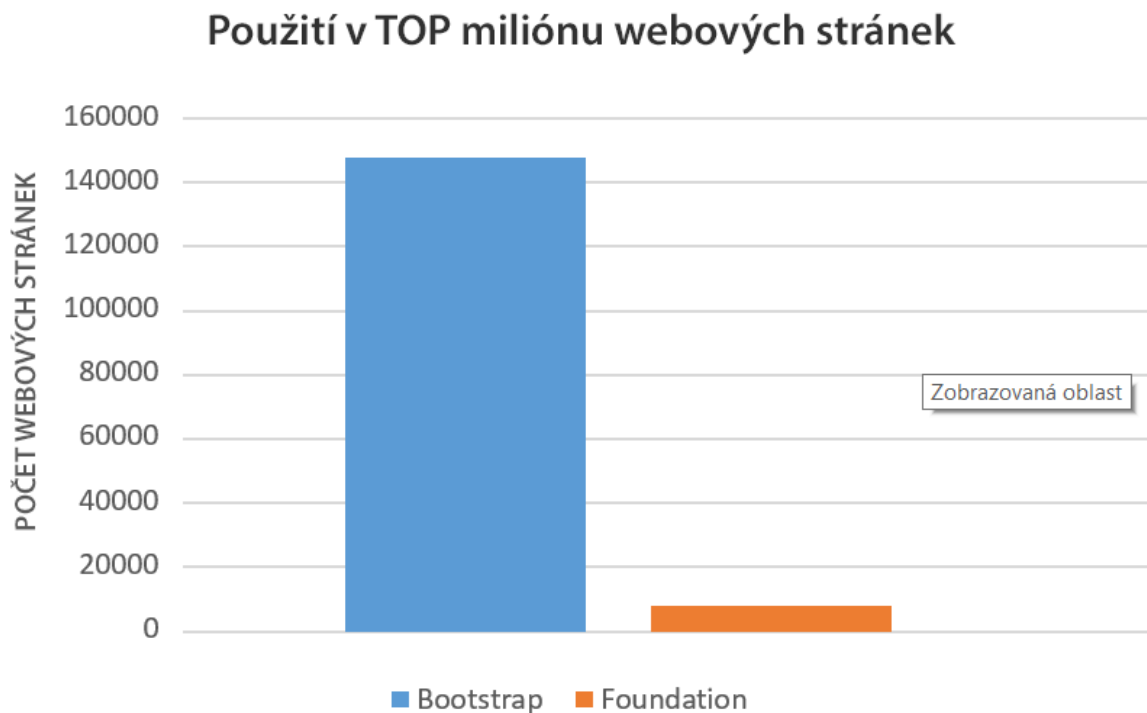
Co se týče srovnání frameworku Bootstrap a Foundation, poskytne nám vcelku názorný obrázek statistika stránky github<sup>12</sup>, která obě tato open-source řešení zastřešuje. Přepis této statistiky můžeme vidět v tabulce 1, ze které lze vyčíst, že mnohem větší část internetové komunity se přiklání právě k Bootstrapu.

	Počet hvězd	Počet forknutí
Bootstrap	94 594	40 920
Foundation	23 048	4 976

Tabulka 1: Přepis statistiky z webu github.

Zdroj dat: <https://github.com/twbs/bootstrap> a <https://github.com/zurb/foundation-sites>

Další statistika, která se přiklání právě k Bootstrapu, je počet použití těchto dvou frameworků u TOP jednoho miliónu nejprestižnějších webových stránek na internetu. Statistika je k vidění na obrázku 15. Výplývá z ní, že z TOP jednoho miliónu stránek, používá právě Bootstrap téměř 148 000 z nich. Naproti tomu druhý jmenovaný Foundation, je použit pouze u 8 000 z nich. Můžeme tedy tvrdit, že Bootstrap ve své kategorii nemá příliš velkou konkurenci.



Obrázek 15: Použití frameworků v TOP jednom miliónu stránek.

Zdroj dat: <http://trends.builtwith.com/docinfo/Twitter-Bootstrap> a  
<http://trends.builtwith.com/docinfo/Foundation>

<sup>12</sup><https://github.com/>

Jelikož práce nemá za úkol vyzdvihnout Bootstrap jako jediné možné a zcela nejlepší řešení, zde je výčet výhody, které považuje autor jako stěžejní:

- Jednoduchost - Bootstrap je vhodný pro všechny vývojáře a designéry i ty méně zkušené. Tento nástroj je vhodné použít pro jednoduché mikrostránky, ale také pro komplexnější aplikace.
- Multiplatformnost - Bootstrap je kompatibilní s většinou dnes moderních prohlížečů, ale také těch starších, a to především i s Internet Explorer 7. Taktéž je to ideální nástroj pro webové projekty určené mobilním zařízením či tabletům.
- Grid systém a responzivita - Umožňuje rozdělit HTML dokument do sloupců (gridu), přičemž podporuje 12 sloupcové rozdělení stránky, pro které jsou předdefinované kaskádové styly, což umožňuje výborné přizpůsobení vzhledu dokumentu velikosti okna internetového prohlížeče.
- Dokumentace - Troufám si tvrdit, že Bootstrap má jednu z nejlepších dokumentací, s jakou jsem se kdy setkal. Dokumentace je velmi přehledná a obsahuje interaktivní ukázky použití a jednoduché vysvětlení bootstrap elementů.
- jQuery pluginy - Bootstrap podporuje interaktivitu s uživatelem prostřednictvím mnoha jQuery pluginů, které obsahuje ve svém jádře. Ty poskytují rozšiřující možnosti uživatelského rozhraní jako dialogové boxy, tooltipy, atd. Také rozšiřují funkcionalitu dosavadních prvků uživatelského rozhraní, to zahrnuje např. funkci auto-doplňování pro vstupní formulářová pole. Při stahování Bootstrap frameworku je možné si nakonfigurovat, které pluginy potřebujeme, aby nebyla knihovna příliš velká a neobsahovala zbytečné věci, které u svého projektu programátor nevyužije.
- Znovu využitelné komponenty - Vedle obvyklých HTML prvků, bootstrap obsahuje další běžně využívané prvky uživatelského rozhraní. To zahrnuje tlačítka s pokročilými možnostmi nastavení a funkcemi (např. seskupování tlačítek nebo tlačítka s drop-down menu, vytváření navigačních listů, horizontální, či vertikální záložky, navigace, drobečková navigace, stránkování, atd.), štítky, pokročilé možnosti typografie, generování náhledů, varovná oznámení a panel průběhu.

#### 4.4.3 Implementace

Kód frameworku je dostupný z oficiálních stránek Bootstrapu<sup>13</sup>. Bootstrap nabízí ke stažení tři verze. První verze obsahuje CSS, JavaScript a fonty a snaží se omezit velikost frameworku na minimum. Druhá verze určená pro práci s LESS obsahuje LESS, CSS, JavaScriptové soubory a dokumentaci. K druhé verzi Bootstrap doporučuje nainstalovat Grunt, který například umožňuje kompilaci LESS a SASS do CSS, minifikaci CSS souborů nebo nahrávání projektů na FTP. Třetí verze je určená pro práci se SASS.

Bootstrap také umožňuje získat zdrojové soubory z CDN (Content Delivery Network)<sup>14</sup> serveru jménem MaxCDN. Výhodou tohoto řešení je úspora datové náročnosti při načítání stránek. Uživatelé, kteří jednou navštíví stránku, které využívá knihoven Bootstrapu, si tyto knihovny stáhnou a při příští návštěvě stránky, využívající technologii Bootstrap, již balíček se zdrojovým kódem nestahují, nýbrž ho získají z Cache.

---

<sup>13</sup><http://getbootstrap.com/>

<sup>14</sup><http://jcas.cz/cdn>



#### 4.4.4 Dokumentace

Dokumentace<sup>15</sup> obsahuje jednoduchý návod s příklady pro první seznámení s frameworkem. Doporučuje také instalaci dalších nástrojů pro správu či kompilaci balíčků. V „*getting-started*“ sekci nalezneme i hotová řešení přímo od Bootstrapu, informace o podporovaných prohlížečích a typ licence, pod kterou je framework zveřejněn. V sekci CSS nalezneme podrobný přehled používaného grid systému, tabulek, formulářů, tlačítek, obrázků a návod pro použití LESS a SASS. V sekci Komponenty se nachází seznam glyphů, použití vstupních polí u formulářů, navigací, vyskakovacích notifikací, progress barů a panelů. Pod sekci JavaScript se skrývají manuály pro tvorbu notifikací v JavaScriptu, webových prezentací, nápověd a dalších funkcí využívajících JavaScript od Bootstrapu.

#### 4.4.5 Customizace

Stránky Bootstrapu podporují customizaci. Jednou z možností je nahrát již předdefinovaný konfigurační soubor a nechat si vygenerovat kód. Druhá možnost spočívá v zaškrtnutí požadovaných funkcí CSS, potřebného seznamu komponent a JavaScriptu. Vývojáři je umožněno nadefinovat proměnné u barev, typografii, odsazení, barvu u jednotlivých komponent, tabulky, barvy tlačítek, formuláře, zlomové body u media queries, samotný grid systém a další nastavení.

### 4.5 WordPress

#### 4.5.1 Co je to WordPress

WordPress je systém pro správu obsahu, zkráceně také CMS (content management system)<sup>16</sup>, který pomáhá bloggerům a webmasterům pravidelně aktualizovat obsah webu bez nutnosti používat tradiční HTML editory jako je například Dreamweaver nebo Frontpage.

Jedná se o open source redakční publikační systém pod GPL licenci. WordPress je nejstahovanější a nejvyhledávanější open source redakční systém na světě. Každý měsíc si ho stáhne téměř milion uživatelů. Dle aktuálních statistik na něm v tuhle chvíli běží přes 74 milionů stránek. Mimo jiné také známá televize CNN nebo noviny The New York Times. WordPress je hojně využíván pro blogy a internetové magazíny, jelikož je stavěn převážně pro tento typ stránek, avšak díky jeho jednoduchosti, otevřeném kódu a vysoké modulárnosti je možno WordPress použít na nespočet jiných „netypických“ stránek.

Samotný redakční systém WordPress je naprogramován ve skriptovacím jazyku PHP a používá databázový systém MySQL. Pro korektní chod všech funkcí je vyžadováno i specifické nastavení serveru v podobě podpory modulu `mod_rewrite`. Některé pluginy pak mají další rozšiřující požadavky; například v podobě nastavení `safe_mode`.

#### 4.5.2 Proč zrovna WordPress

WordPress je jedním z nejoblíbenějších redakčních systémů a na světě jej momentálně používá kolem 76 500 000 stránek<sup>17</sup>. Tato statistika jasně ukazuje, že jde o velmi populární CMS. Toto tvrzení také potvrzuje porovnání vyhledávacích frází v čase pomocí nástroje třetí strany Google Trends, kde jasně převyšuje svou konkurenci právě WordPress, viz graf 16

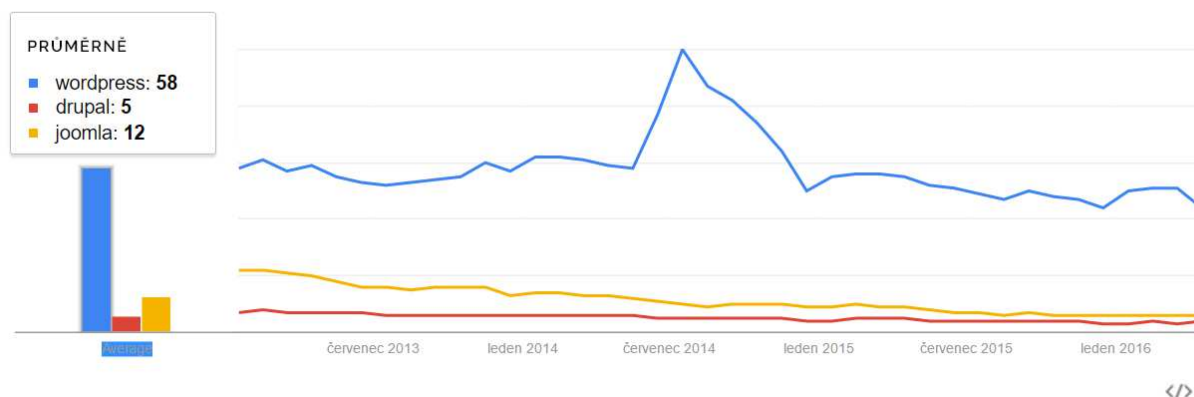
---

<sup>15</sup><http://getbootstrap.com/css/>

<sup>16</sup><http://www.adaptic.cz/znalosti/slovnicek/cms>

<sup>17</sup><http://expandedramblings.com/index.php/wordpress-statistics/>

## Zájem vyhledávání v průběhu času

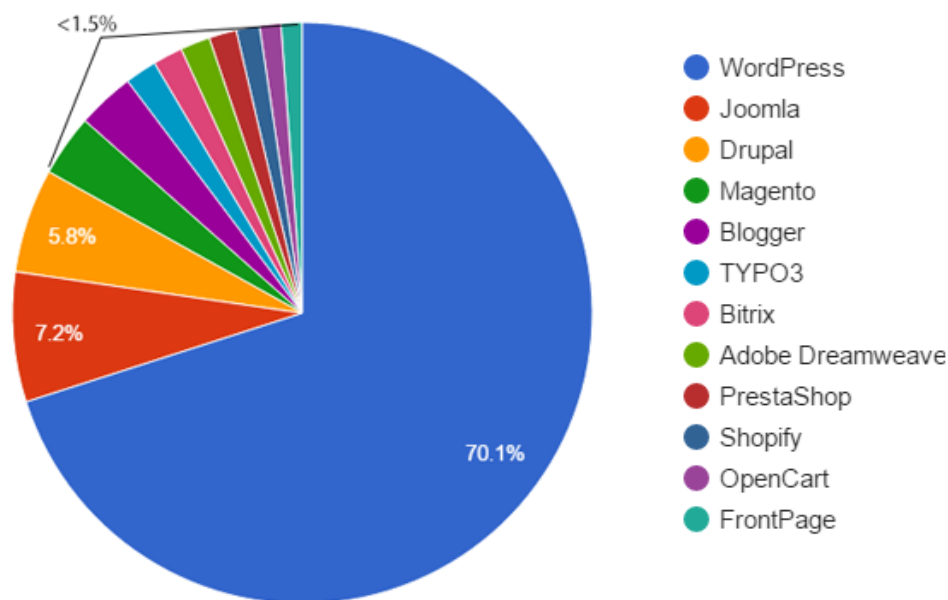


Obrázek 16: Popularita systému WordPress a jeho největších konkurentů podle vyhledání v čase.

Zdroj: <https://www.google.cz/trends>

Další ze statistik, která dokazuje oblíbenost systému WordPress, ukazuje procentní podíl jednotlivých redakčních systémů, které jsou použity na webových stránkách. Nutno podotknout, že ve statistikách nejsou zahrnuty webové stránky, které nepoužívají žádný redakční systém, nebo stránky, které běží na on-line systémech typu Webnode, Tumblr, a jiné. Statistiku může vidět níže, na obrázku 17. Podíl WordPressu zde dosahuje obrovských 70%. Náskok na své konkurenty má tedy obrovský.

## Použití redakčního systému



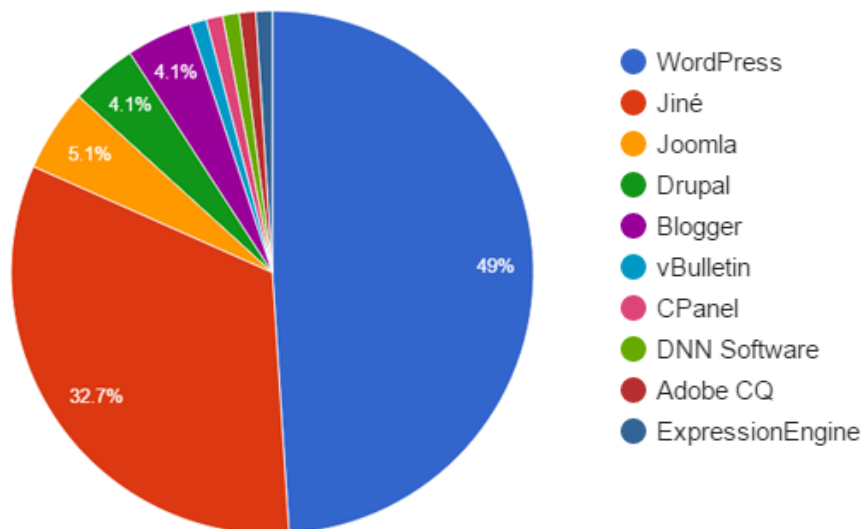
Obrázek 17: Procenta použití redakčního systému.

Zdroj:[http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)

Poslední statistika, která poukazuje na oblíbenost WordPressu, je vidět na obrázku 18. Statistika ukazuje nasazení redakčních systémů u velkým a známých webů. Graf reprezentuje podíl WordPressu v TOP jednom milionu stránek na světě, a porovnává ho s nejčastěji používanými systémy na těchto stránkách. Jak můžeme vidět, podíl WordPressu je úctyhodných 49% a své konkurenty tak opět válkuje. Další velkou část, přesněji 32.7%, zabírají systémy šité na míru, které nejsou nikterak volně dostupné. Tato statistika jasně dokazuje, že známé weby upřednostňují právě WordPress<sup>18</sup>.

<sup>18</sup>Údaje byly naměřeny k Únoru 2016

## Použití redakčního systému v TOP milionu stránek



Obrázek 18: Použití redakčního systému v TOP milionu stránek.

Zdroj: <http://trends.builtwith.com/cms>

Jak vyplynulo z naměřených statistik, WordPress je opravdu jeden z nejoblíbenějších redakčních systémů, které jsou volně dostupné na trhu. Ačkoliv si práce neukládá za cíl obhajovat WordPress jako nejlepší řešení, vypíšu zde pár důvodů, proč jsem si jej zvolil právě já:

- S oblíbeností WordPressu souvisí existující obrovská komunita vývojářů a designerů.
- Pro WordPress nalezneme obrovské množství pluginů. Aktuálně existuje přibližně 43 tisíc pluginů<sup>19</sup>
- Pro WordPress také nalezneme velké množství vzhledů, jak volně ke stažení, tak komerčních.
- I přesto, že byl WordPress nejprve vyvíjen jako blogovací systém, dnes nabízí obrovské možnosti použití: blog, fotogalerie, portfolio, firemní prezentace, e-shop, magazín, atd. To vše díky jeho modularitě a také množstvím hotových rozšíření.
- WordPress disponuje masivní komunitou, z čehož vyplývá i dobrá zpětná vazba a podpora ze strany této komunity. Existuje nespočet diskuzí či fór, které se WordPressu úzce věnují.

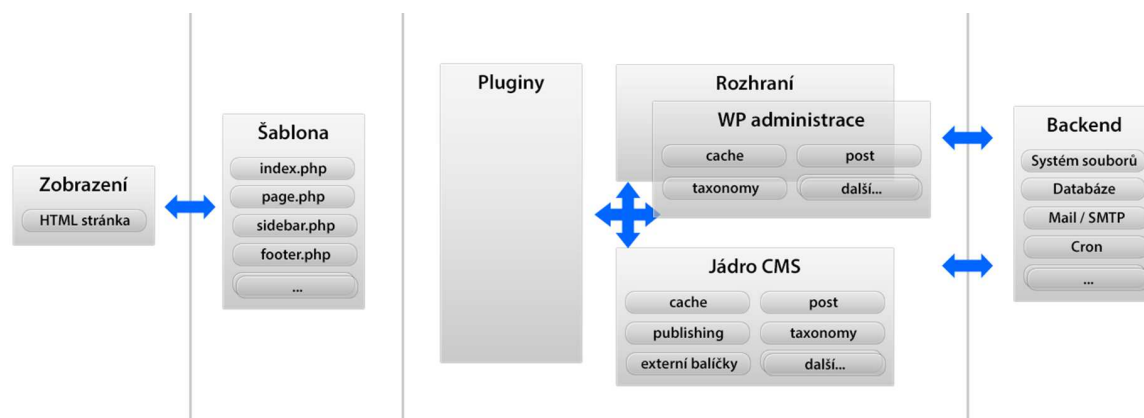
<sup>19</sup><https://wordpress.org/plugins/>

- Velkou výhodou vidím ve srozumitelně napsané dokumentaci celého systému, která je k dispozici on-line a je velmi uživatelsky přívětivá.
- WordPress se chlubí, že je tzv. „SEO friendly“, což znamená, že je velmi dobře optimalizován pro vyhledávače.
- Velmi snadná a rychlá instalace systému. Jsme schopni web spustit do 5 minut.

### 4.5.3 Architektura WordPressu

WordPress má klasickou stavbu redakčního systému. Pokud má uživatel dostatečná oprávnění, má přístup do jeho administrace, tedy do tzv. „backendu systému“, čímž může vkládat a editovat obsah a jinak upravovat finální podobu webové stránky postavené na WordPressu. V popředí systému stojí samotný „frontend“ aplikace, který vidí obyčejní návštěvníci webu, a jejichž práva jsou limitována pouze na možnosti číst obsah, komentovat, nebo jiné možnosti, dle našich zvolených kritérií. Za těmito dvěma vrstvami stojí technické pozadí v podobě samotného jádra WordPress, pod níž věží serverové technologie a hardware.

Základní struktura WordPressu obsahuje několik samostatných částí, které spolu komunikují. Jedná se o jádro systému, administraci systému a uživatelské soubory. Tyto části jsou zastoupeny složkami *wp-admin* (soubory administrace), *wp-content* (uživatelské soubory) a *wp-includes* (jádro systému). Jakým způsobem spolu tyto části komunikují můžeme vidět na zjednodušeném diagramu architektury, viz obrázek 19. Mnohem podrobnější digram architektury WordPressu je součástí Přílohy A.



Obrázek 19: Zjednodušená komunikace jednotlivých částí WordPressu.

Jak můžeme na zjednodušeném diagramu vidět zprava, nejprve máme zde výpis technologií, na kterých běží WordPress. Jedná se vlastně o technické zázemí celého systému. Ty nejzásadnější jsou především hostingové úložiště s podporou databázového úložiště, souborový systém, mailový server a cron úlohy.

S tímto technickým zázemím dále komunikuje jádro systému, administrace systému a pluginy. Nad administrací systému se ještě nachází samotné rozhraní, které slouží ke komunikaci s HTTP, XML-RPC, AtomPub a jinými kanály. Samotné jádro se skládá z jednotlivých balíčků, které se starají o celkovou funkcionalitu systému jako, vkládání příspěvků, práci s pluginy, zálohování, atd. Mezi tyto balíčky jádra mohou být přidány externě dodané pluginy, které se postarají o rozšíření celkové

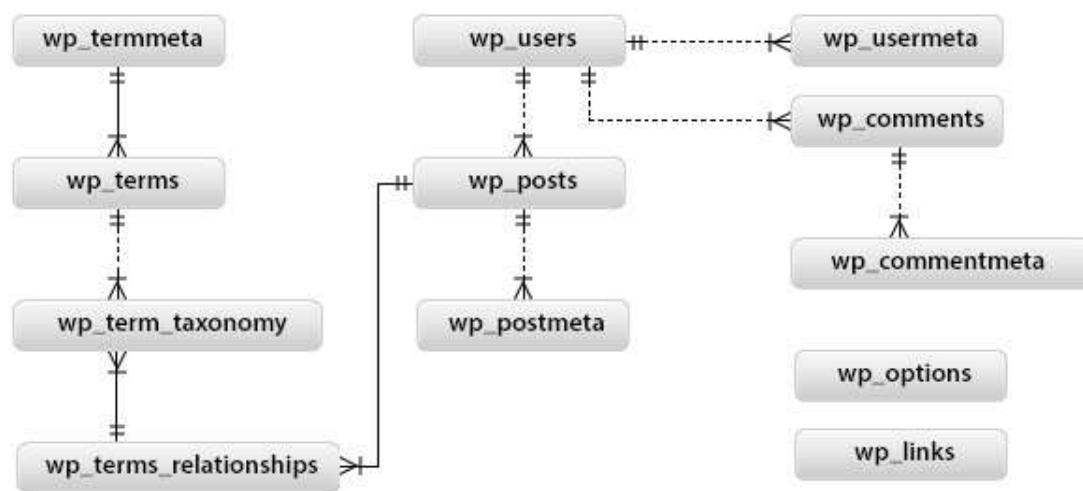
funkcionality systému. Díky možnosti zasahovat do jádra systému právě externími balíčky, můžou vývojáři ovlivnit běh své webové stránky právě externímu pluginu a nemusejí tedy zasahovat přímo do zdrojových souborů jádra a administrace. Tyto přímé zásahy do zdrojových souborů jsou ze strany silně nedoporučovány, jelikož hrozí ztráta podpory a oficiálních aktualizací ze strany WordPressu.

Nad jádrem samotného systému a administrace stojí dodávané šablony vzhledu, které se starají o samotné podání obsahu uživatelům v námi zvoleném grafickém podání. I z této části systému můžeme přistupovat do jádra systému a měnit jeho funkcionalitu (soubor `functions.php`), například pro jiný formát obsahu vráceného z databázového systému. Šablona vzhledu se skládá s několika souborů, více o nich v kapitole 6.2.

Poslední část architektury je výsledný vygenerovaný výstup na základě našeho nastavení ve formě samotné HTML stránky. Tento výstup se skládá ze souboru HTML, stylů CSS a Javascriptových knihoven.

#### 4.5.4 Databázový model WordPressu

Model databázového systému výchozí čisté instalace můžeme vidět na obrázku 20. WordPress běží na MySQL databázi, která je jeho hlavním úložištěm veškerého obsahu. Najdeme se veškerý obsah (krom mediálních souborů), který WordPress nějakým způsobem potřebuje uložit, jako nastavení, informace o uživateli, příspěvky, kategorie, komentáře a mnoho dalších. Při výchozí instalaci nalezneme v databázi 11 základních tabulek, viz tabulka 2. Tento počet může narůstat, především s přibývajícími pluginy, které svůj obsah mohou do této databáze ukládat také a můžou si také vytvářet vlastní tabulky. Kompletní a podrobnější datový model WordPressu je součástí Přílohy B.



Obrázek 20: Datový model systému WordPress.

Název Tabulky	Popis Tabulky
wp_commentmeta	Obsahuje meta data patřící komentářům (datum, čas, autor).
wp_commts	Obsahuje obsah samotného komentáře.
wp_links	Obsahuje odkazy používané v postranním panelu webu.
wp_options	Obsahuje nastavení systému provedené v administraci.
wp_posts	Obsahuje příspěvky, stránky a také menu.
wp_term_relationship	Zde jsou ukládány vazby mezi příspěvkem a jeho kategoriemi, štítky, atd.
wp_term_taxonomy	Obsahuje hierarchické uspořádání kategorií, odkazů a štítků.
wp_termmeta	Obsahuje meta data pro kategorie, odkazy a štítky.
wp_terms	Zde jsou uloženy samotné kategorie a štítky.
wp_usermeta	Obsahuje meta data pro uživatele.
wp_users	Zde jsou shromažďovány všechny uživatelské profily.

Tabulka 2: Tabulky databázového modelu výchozí instalace.

## 5 Bootstrap v praxi

### 5.1 Implementace Bootstrapu

Bootstrap lze v základu získat a použít velmi snadno. Navštívíme oficiální web Bootstrapu<sup>20</sup>, kde si celý framework můžeme zdarma stáhnout. Máme dvě možnosti jak toho docílit:

1. Stáhneme si soubory Bootstrapu - tlačítko Download Bootstrap. Zde máme samozřejmě na výběr ze třech balíčků zmiňovaných v kapitole 4.4.3. Po stažení dostaneme balík obsahující minimalizované soubory kaskádových stylů a JavaScriptu. Tahle varianta je vhodná, pokud chceme mít větší kontrolu na implementovanými soubory, jelikož je máme fyzicky k dispozici.
2. Natáhnutí zdrojových souborů z CDN - do své aplikace nebo www stránky vložíme odkazy na soubory Bootstrapu uložené na CDN serverech. Tato metoda je jednodušší, a také zmírňuje zátěž načítání zdrojových souborů při zobrazení webové stránky.

Pro naše řešení postačí druhá možnost načtení Bootstrapu, jelikož neplánujeme zasahovat do jeho vnitřní struktury jako takové. Nejprve si načteme soubory s Bootstrap kaskádovými styly, ve kterém najdeme nastýlované všechny Bootstrap komponenty, které můžeme poté v HTML dokumentu používat. Dále si můžeme načíst volitelné Bootstrap CSS pro již nastýlovaný vzhled jednotlivých základních prvků HTML, jako jsou tlačítka, tabulky, seznamy, typografie a další. Poté ještě přiložíme CSS soubor se styly frameworku FontAwesome, aby jsme mohli jednoduše používat jeho nastýlované ikony. A jako poslední přiložíme náš vlastní CSS soubor, který se bude starat o naše finální stylování HTML dokumentu. Ujistěte se, že váš CSS soubor voláte až jako poslední, a to zejména proto, abychom se ujistili, že naše styly vždy přepíšou styly nadřazené. Pokud bychom chtěli přestylovat HTML komponentu, která je již součástí Bootstrap CSS, jednoduše ji nastylujeme v našem vlastním CSS a tento styl bude následně na komponentu aplikován, bez ohledu na to, že tato komponenta již svůj Bootstrap styl má.

Dále si načteme soubory s JavaScriptem, abychom mohli využívat všech Bootstrap pluginů a funkcí. Také si načteme knihovnu jQuery, která je nepostradatelná pro samotné fungování tohoto frameworku. A jako poslední si načteme soubor s vlastními JavaScriptovými funkcemi, a to ze stejného důvodu, který jsme si vysvětlili u vlastního CSS souboru. Tyto JavaScriptové soubory jsou doporučeny vložit před koncovou značku tagu `</body>`

Finální šablona se všemi potřebnými soubory pro správnou funkci Bootstrapu, a ze které budeme vycházet při tvorbě každé nové podstránky našeho webu bude tedy vypadat jako na obrázku 21.

---

<sup>20</sup><http://getbootstrap.com/getting-started/>



```

<!DOCTYPE html>
<html lang="cs">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Bakalarska prace</title>

  <!-- Poslední verze Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">

  <!-- Volitelný vzhled (nepovinné) -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-theme.min.css">

  <!-- Poslední verze FontAwesome CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css">

  <!-- Vlasni stylyl - NECHAT NA KONCI -->
  <link href="assets/css/custom.css" rel="stylesheet">
</head>
<body>

  <!-- ZDE VLOŽ SVŮJ OBSAH -->

  <!-- KONEC OBSAHU -->

  <!-- Pripojeni JQuery knihovny -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>

  <!-- Připojení Bootstrap JavaScriptu-->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

  <!-- Připojení vlastního JavaScriptu-->
  <script src="assets/js/custom.js"></script>
</body>
</html>

```

Obrázek 21: Základní šablona s načtenými soubory

## 5.2 Hlavní pilíř Bootstrapu - Grid systém

Jak již bylo zmíněno výše, vlastnost flexibilního designu je v Bootstrapu zajištěna pomocí grid systému<sup>21</sup>, který je stavebním prvkem celého Bootstrapu. Tento grid systém rozděluje šířku celého výřezu aktuálně používané obrazovky do 12ti abstraktních sloupců, z nichž má každý šířku 8,33%. Základním prvkem celého layoutu je obalovací třída *.container* pro fixní šířku layoutu nebo *.container-fluid* pro šířku plně přizpůsobující se maximální možné velikosti. Dalším stavebním prvkem je element se třídou *.row*, dále jen řádek, který se vždy roztáhne na 100% prvku, do kterého je vnořen a vytvoří tak horizontální skupinu 12ti abstraktních sloupců. Kýžený layout tvoří skupiny sloupcových buněk seskupených do jednotlivých řádků. Principiálně je to velmi podobné jako u klasických HTML tabulek. Prezantovaný obsah by měl být vkládán pouze do sloupcových buněk, a jen ty by měly být potomky řádku.

Šířka vytvářeného sloupce je dána vždy pomocí CSS třídy a to v závislosti na tom, kolik jednotek má obsáhnout. Formát sloupcové třídy je *.col-X-Y*, kde *X* je šířka celkového výřezu, při kterém nastavení platí neboli maximální rozlišení zobrazovací obrazovky zařízení, a *Y* je počet jednotek

<sup>21</sup><https://getbootstrap.com/examples/grid/>

respektive sloupců, které má element v celém výřezu z 12ti sloupců zaplnit. Dostupné šířky výřezů jsou:

- xs - Velmi malé zařízení, jako např. mobilní telefony ( <768px )
- sm - Menší zařízení, jako např. tablety( >=768px )
- md - Střední zařízení, např. přenosné počítače a tablety( >=992px )
- lg - Větší zařízení, většinou stolní počítače ( >=1200px )

Tyto třídy je vhodné spolu kombinovat, čímž docílíme různou šířku panelů v závislosti na velikosti aktuálního zobrazovacího zařízení. Jako příklad nám poslouží div s Bootstrap třídou `.col-xs-12` `.col-md-6`. Při rozlišení menším než 768 (parametr xs) se bude tento div roztahovat přes celou šířku aktuální zobrazovací plochy, přesněji bude zabírat všech 12 sloupců z 12ti možných. Překročí-li rozlišení obrazovky hodnotu 992px (parametr md), bude tento div zabírat pouze polovinu zobrazovací plochy, přesněji bude zabírat pouze 6 sloupců z 12ti možných.

V případě, že budeme potřebovat element posunout doprava, tedy chceme element, který nebude vykreslen od prvního sloupce, použijeme třídu `.offset-X-Y`, kde *X* je opět šířka výřezu, a *Y* je počet sloupců, které se mají přeskočit. Tyto třídy můžeme opět společně kombinovat. Ukázkou grid systému můžete vidět níže na obrázku 22.

Poslední pomůckou gridového systému je možnost zobrazení rozdílného layoutu pro daná zařízení. Díky přednastaveným třídám `.visible-X`, respektive `.hidden-X`, kde *X* je opět parametr šířky výřezu (xs, sm, md, lg), můžeme definovat, jaký prvek se má v layoutu zobrazit či nezobrazit pro konkrétní rozlišení. Můžeme tedy nastavit, aby se obsah zobrazoval (třída visible) nebo nezobrazoval (třída hidden) na určeném rozlišení dle paramteru *X*.



Obrázek 22: Ukázka systému Grid

### 5.3 Přípravené styly

Velkou výhodou tohoto CSS frameworku jsou bezpochyby defaultně předchystané kaskádové styly CSS. Pokud nepotřebujeme originální design našich webových stránek, využití těchto přednastavených stylů naší práci značně urychlí. Stačí tedy využívat definované CSS styly, které jsou již otestovány pro kompatibilitu ve všech zařízeních a také pro mobilní telefony. V následujících podkapitolách si tedy ukážeme některé ze základních předdefinovaných kaskádových stylů Bootstrapu. Jelikož CSS knihovna Bootstrap stylů je velmi obsáhlá, nebudou přiblíženy všechny dostupné třídy. Kompletní výpis všech dostupných tříd, jejich popis, použití a také vzorové příklady naleznete v dokumentaci Bootstrapu<sup>22</sup>.

#### 5.3.1 Typografie

Ve výchozím nastavení používá Bootstrap písma *Helvetica Neue*, *Arial* a *Sans-serif*. Tato písma jsou považována za webový standart a je tedy málo pravděpodobné, že by některé koncové zařízení těmito fonty nedisponovala. V případě, že ani jedno z těchto písem není na zařízení dostupné, je použito základní písmo zařízení. Výchozí globální nastavení je 14px velikost písma s 1.428em mezerou mezi řádky. Bootstrap nabízí automaticky předdefinované všechny úrovně nadpisů (h1 - h6), popřípadě jsou tyto styly dostupné přes třídu **.h1** - **.h6**. Ukázku předdefinovaných nadpisů můžete vidět na obrázku 23

<sup>22</sup><http://getbootstrap.com/css/>

h1. Bootstrap heading	Semibold 36px
h2. Bootstrap heading	Semibold 30px
h3. Bootstrap heading	Semibold 24px
h4. Bootstrap heading	Semibold 18px
h5. Bootstrap heading	Semibold 14px
h6. Bootstrap heading	Semibold 12px

Obrázek 23: Ukázka nastýlovaných nadpisů všech úrovní.  
Zdroj: <http://getbootstrap.com/css/#type>

Pro potřeby typografie Bootstrap definuje několik svých nových HTML značek, jako například:

- `<mark>` - Text v tomto bloku bude vůči okolnímu textu zvýrazněn
- `<del>` - Text v tomto bloku bude přeškrtnut
- `<ins>` - Text v tomto bloku zvýrazněn jako dodatek
- `<small>` - Text v tomto bloku bude mít 85% velikost písma, než-li jeho rodič.

Dále CSS styly Bootstrapu nabízí několik tříd pro zarovnání textu, transformování textu, seznam položek, další různá zvýraznění pro adresy, citace popisky, a mnoho dalších tříd, které souvisejí právě s typografií. Podrobnější výpis tříd naleznete v dokumentaci pod sekci Typography<sup>23</sup>. Bootstrap také obsahuje sadu 250 ikon, které poslouží k doplnění tlačítek, navigace, odkazu o vizuální prvky. Seznam těchto ikon je součástí komponenty zvané Glyphicons<sup>24</sup>, na jejichž stránce najdeme kompletní výpis ikon, spolu s návodem na jejich použití.

### 5.3.2 Tabulky

Pro nastýlování tabulek Bootstrap nabízí třídu `.table`, která styluje tabulku bez okrajů, popřípadě třídu `.table-bordered`, která styluje tabulku s okraji. Pokud chceme zapnout zvýrazňování aktuálních řádků, poslouží nám třída `.table-hover`. Popřípadě třída `.table-striped`, která pomocí CSS selektoru `:nth-child` docílí u tabulky tzv. zebrování. Tyto třídy mezi sebou můžeme samozřejmě různě kombinovat. Ukázku takovéto tabulky najdeme na obrázku 24. Pokud si nepřejeme, aby byla tabulka zcela responzivní, obalíme kteroukoliv tabulku elementem s třídou `.table-responsive`, tato tabulka se bude na malých zařízeních (pod 768px) zobrazovat s horizontálním posuvníkem.

<sup>23</sup><http://getbootstrap.com/css/#type>

<sup>24</sup><http://getbootstrap.com/components/#glyphicons>

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```

<table class="table table-striped">
  ...
</table>

```

Obrázek 24: Ukázka použití nastýlovaných tabulek.

Zdroj: <http://getbootstrap.com/>

Pomocí předdefinovaných stylů můžeme také barevně odlišit různé buňky, či řádky. Přehled těchto tříd naleznete v tabulce 3.

Třída	Popis chování
.active	Aplikuje na buňku či řádek barvu zvýraznění, defaultně světle modrá barva.
.success	Indikuje úspěšnou či pozitivní akci, defaultně zelená barva.
.info	Indikuje neutrální změnu nebo akci, defaultně bledě modrá barva
.warning	Indikuje varovnou akci, defaultně světlé oranžová barva
.danger	Indikuje nebezpečnou či negativní akci, defaultně červená barva

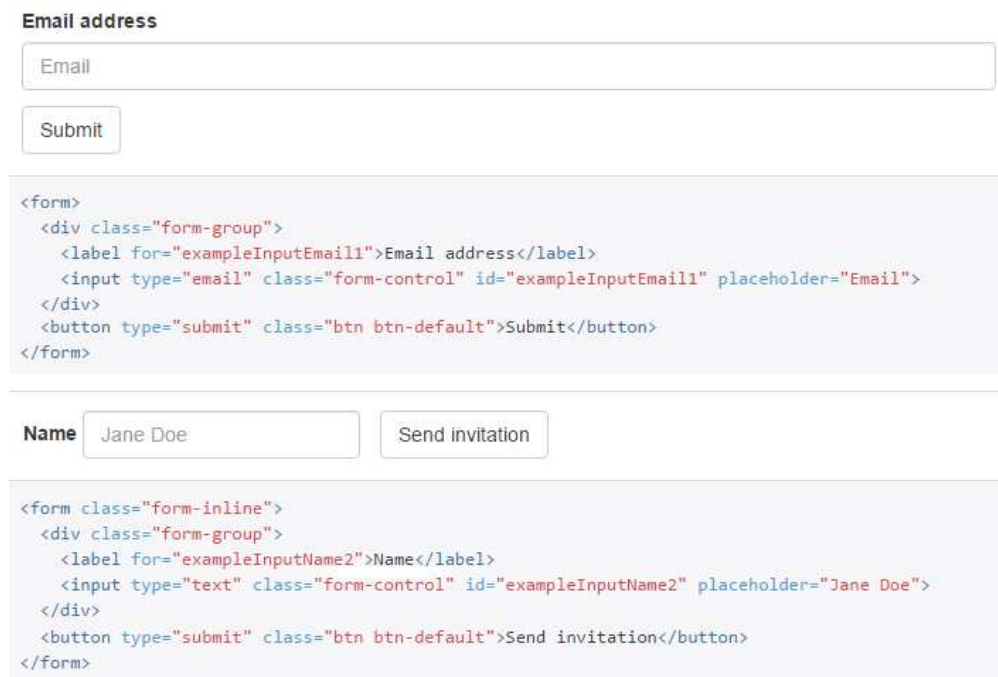
Tabulka 3: Seznam tříd pro barevné odlišení buňek či řádků tabulky.

Kompletní výčet a popis tříd pro formátování tabulek naleznete v dokumentaci v sekci Tables<sup>25</sup>.

### 5.3.3 Formuláře

Prvky formulářů jsou v Bootstrapu defaultně nastaveny jako blokové, tudíž se zarovnávají pod sebou a položky formuláře mají nastavenou šířku jako 100%. Jednotlivé položky jsou dostupné pod obalovou třídou **.form-control**, která seskupuje popis a vstupní pole do jednoho logického celku pro optimální formátování a odsazení. Pokud chceme formulář zobrazit do jedno řádku se zarovnáním vlevo, tedy neblokové zobrazení, můžeme využít třídu **.form-inline**, která se aplikuje přímo na značku **<form>**. Prvky v tomto formuláři budou mít nyní svou šířku nastavenou jako **auto**. Výchozí styl formulářů, spolu se zdrojovým kódem, je názorně ukázán na obrázku 25.

<sup>25</sup><http://getbootstrap.com/css/#tables>



Obrázek 25: Ukázka použití nastýlovaných formulářů.

Zdroj: <http://getbootstrap.com/>

Bootstrap dále styluje většinu základních typů formulářových prvků. Pro funkčnost je důležité mít správně vyplněný typ formulářového prvku, dle dokumentace. Bootstrap podporuje tyto HTML5 typy formulářových prvků: *text*, *password*, *datetime*, *datetime-local*, *date*, *month*, *time*, *week*, *number*, *email*, *url*, *search*, *tel*, a *color*.

Kompletní výpis všech formulářových stylů s návodem na použití najdeme v dokumentaci pod kapitolou Forms<sup>26</sup>.

#### 5.3.4 Tlačítka

Třídy nachystané pro stylování tlačítek jsou určena pro tagy `<a>`, `<button>` nebo `<input>`. Bootstrap nám nabízí celkem 7 odlišných tříd pro různá tlačítka. Základní třída, která určuje, že se jedná o tlačítko, je třída `.btn`. Po této třídě dále následuje definice stylů tlačítka, jejichž výčet a použití můžete vidět na obrázku 26. Můžeme také určit velikost tlačítka a tím tedy určit významnost tohoto tlačítka. K dispozici jsou třídy `.btn-lg` (velké tlačítko), `.btn-sm` (malé tlačítko) a `.btn-xs` (extra malé tlačítko). Kompletní výpis všech stylů tlačítek a jejich použití naleznete v dokumentaci v kapitole Buttons<sup>27</sup>.

<sup>26</sup><http://getbootstrap.com/css/#forms>

<sup>27</sup><http://getbootstrap.com/css/#buttons>



Obrázek 26: Ukázka použití nastýlovaných tlačítek.

Zdroj: <http://getbootstrap.com/>

## 5.4 Komponenty

Vedle obvyklých HTML prvků, Bootstrap obsahuje další běžně využívané prvky uživatelského rozhraní. To zahrnuje tlačítka s pokročilými možnostmi nastavení a funkcemi. Unikátní komponentou je zde například Jumbotron, jenž slouží pro zvýraznění důležitého informačního obsahu. Framework nabízí také poměrně velké množství komponent, které jsou JavaScriptového typu. Z nich je velmi využitelná komponenta Modal, díky které je možné do stránky zakomponovat vyskakovací okna s libovolným obsahem. Další z nich velmi dobře využitelná je komponenta Tab, pomocí které je možné rozdělit obsah do tematických záložek. Oblíbenou komponentou je Carousel, do které je možné umístit sadu obrázků s popisem, které se v nastaveném časovém intervalu obměňují. Část těchto komponent si popíšeme v následujících podkapitolách. Další ukázka vybraných komponent je poté součástí Přílohy C. Kompletní výpis komponent společně s návodem na jejich použití naleznete v příslušné části Bootstrap dokumentace<sup>28</sup>.

### 5.4.1 Glyphicons

O této komponentě jsem již mluvil v kapitole 5.3.1. Jedná se o sadu předpřipravených grafických ikon. Ikon je v Bootstrapu téměř na 260. Při jejich používání se nedoporučuje kombinovat ikonky přímo s jinými komponentami. Namísto toho je vhodné vytvořit formátovací značku `span` s třídou `glyphicon` a druhou třídou vyjadřující danou ikonku. Dokumentace dále doporučuje, abychom při používání ikon, které mají dekorativní funkci, přidali atribut `aria-hidden="true"`. Ten zabraňuje nechtěnému a matoucímu výstupu u asistivních technologií.

### 5.4.2 Rozbalovací nabídka

Tuto komponentu řeší framework přes třídu `.dropdown` umístěnou v `divu`. Přidáním tlačítka s ikonkou šipky a seznamem s třídou `.dropdown-menu` vytvoříme jednoduché výsuvné menu. Do komponenty můžeme také vložit nadpis, ten vložíme do seznamu s třídou `.dropdown-header`. Framework řeší i oddělovače, třída `.divider`.

---

<sup>28</sup><http://getbootstrap.com/components/>

### 5.4.3 Skupina tlačítek

Pomocí frameworku můžeme seskupovat jednotlivá tlačítka do přehledných panelů. Je vhodné do obalujícího divu určit roli, aby byla tlačítka ve skupině interpretována správně. Dokumentace frameworku dále uvádí, že je vhodné odlišit skupinu pomocí `aria-label`. Doporučení se týká hlavně při využívání různých asistivních technologií. Velikost panelu nastavujeme v divu pomocí tříd `.btn-group-lg`, `.btn-group-sm` a `.btngroup-xs`. Defaultně je panel horizontální, vertikální rozložení nám umožní třída `.btngroup-vertical`, kterou nahradíme `.btn-group`.

### 5.4.4 Navigační lišta

Navigační lišta v Bootstrapu může obsahovat další již zmíněné komponenty. Potřebným minimem je tag `nav` s třídou `.navbar` a `.navbar-default`. Poté se vytvoří hlavička navigace, která umožňuje v případě mobilních zařízení využívat výsuvného menu a nakonec vlastní menu. Tato komponenta vyžaduje nainportovaný JavaScriptový plugin pro mobilní navigaci. Do navigační lišty můžeme umístit také logo společnosti za pomoci třídy `.navbar-brand`, formuláře s třídou `.navbar-form`, tlačítka `.navbar-btn` a jiné komponenty. K určení umístění navigační lišty slouží třída `.navbar-fixed-*`, kde uvedeme buď `top` (umístění nahoře), či `bottom` (umístění dole). Pokud chceme statické menu, využijeme třídy `.navbar-static-top`.

### 5.4.5 Progress bar

Komponentu vytvoříme kombinací dvou divů, jednoho s třídou `.progress` a druhou s třídou `.progress-bar`. Procentní vyjádření vkládáme do divu s třídou `.progress-bar`. Komponentu můžeme stylovat pomocí třídy `.progress-bar-X` (kde `X` může být `success`, `info`, `warning` nebo `danger`). Můžeme přidat také různé animace a další efekty, ale ty nejsou podporovány ve všech prohlížečích. Progress bary nejsou funkční v IE 9 a starším.

## 5.5 Přizpůsobení Bootstrapu

Úprava stylů, typografie, obsažených komponent a dalších je v Bootstrapu poměrně jednoduchá. Dochází k ní již před stažením samotného frameworku do počítače. Pro přizpůsobení je dostupná speciální webová stránka<sup>29</sup>, kde je možné přenastavit všechny důležité parametry. Je možné také upravit proměnné atributy pro preprocesor LESS. V případě, že uživatel není webdesignér a nechce zdoluhavým nastavováním ztrácet čas, může využít nějakou z přednastavených šablon. Dobrým zdrojem šablon je například webová stránka Bootswatch<sup>30</sup>. Přímou na webové stránce frameworku je také dostupných několik základních šablon s příklady různých druhů layoutů<sup>31</sup>. Pro další úpravy je již nutné vytvořit svojí vlastní CSS šablonu.

## 5.6 Tvorba webu pomocí Bootstrapu

Tato podkapitola nás zasvětila do rychlé tvorby webové stránky pomocí frameworku Bootstrap s využitím předpřipravených stylů, které jsme si popsali v kapitole 5.3. Bude se jednat o jednoduchou webovou stránku, na které si demonstrujeme rychlost tvorby responzivní webové stránky pomocí

---

<sup>29</sup><http://getbootstrap.com/customize/>

<sup>30</sup><https://bootswatch.com/>

<sup>31</sup><http://themes.getbootstrap.com/>



předpřipravených stylů. Stránka bude také sloužit jako demonstrace základních Bootstrap komponent v praxi, jako je tvorba horizontální a vertikální navigace, typografie, použití ikon a další. Tato podkapitola počítá se znalostí předešlé podkapitoly 5.2, tudíž k vysvětlení jednotlivých Bootstrap tříd gridovacího systému se vracet nebudeme.

V hlavičce webové stránky si načteme všechny potřebné soubory frameworku, jako je ukázáno v předchozí kapitole 5.1. Rozložení našeho ukázkového webu bude klasické, a bude sestávat s hlavičky, která ponese logo stránky, název stránky a doplňující popis stránky. Pod hlavičkou najdeme hlavní horizontální navigaci, která bude podporovat vnořené submenu. Obsahovou kostru budou tvořit dva sloupce. Levý, pro vedlejší vertikální navigaci, která může sloužit pro seznam rubrik, tagů atd. A pravý sloupec, který bude sloužit pro samotný obsah. Na samotném konci stránky najdeme klasickou patičku, obsahující dodatečné či copyright informace o samotné stránce.

Hlavička webu je obalena HTML5 tagem `<header>`, čistě kvůli HTML5 sémantice, která říká, že hlavní hlavička webu, či hlavička samostatné sekce, by měla být do tohoto tagu vnořena. Dále použijeme Bootstrap třídu `.container`, jelikož chceme logo a text hlavičky odsadit od samotného okraje stránky. Do této třídy zanoříme element s Bootstrap třídou `.row`, který dále nebude rozdělovat pomocí grid systému, jelikož požadujeme, aby se hlavička chovala vždy a všude stejně. Tedy přesněji, aby hlavička vždy zabírala 12 možných pomyslných sloupců. Naše logo stránky je zaobaleno Bootstrap třídou `.navbar-brand`, což je třída, která obrázek optimalizuje pro všechny mobilní zařízení. Využijeme ji tehdy, pokud máme logo ve vysokém rozlišení a necháme na Bootstrapu, ať si obrázek sám zkomprimuje na požadovanou velikost. Docílíme tak kvalitnímu a ostrému zobrazení loga na každém rozlišení<sup>32</sup>. Pro titulek stránky použijeme vlastní definovanou třídu, pouze s estetických důvodů, tedy možnosti vlastního naformátování úvodního textu.

Pro hlavní horizontální menu využijeme Bootstrap komponentu pro webovou navigaci. Celé menu je vnořeno do třídy `.navbar`, což je třída udržující předpřipravený styl pro rychlou tvorbu horizontálních a vertikálních navigací. Výchozí navigace se nese ve stylu bílého pozadí a černého písma, pro účely naší vzorové stránky je však vyžadováno menu inverzní, tedy černé pozadí a bílé písmo. K tomuto účelu zde najdeme Bootstrap třídu `.navbar-inverse`, kterou obalovému elementu navigace přidáme. V neposlední řadě můžeme definovat samotné umístění navigace, Bootstrap nám nabízí tři možnosti umístění navigace. Fixní na vrcholu stránky, třída `.navbar-fixed-top`, které se bude posouvat se samotnou stránkou. Statické menu, jako v našem případě třída `.navbar-static-top`, které chová jako klasický blokový prvek. A menu fixní na spodku stránky, třída `.navbar-fixed-bottom`, které bude při prvním náhledu a posunu stránky neustále umístěna ve spodku stránky. Další užitečná třída je třída `.navbar-toggle`, která udržuje vzhled menu po tom, co zobrazovací plocha nedosahuje rozlišení 768px na šířku. Laicky řečeno, elementy uvnitř udávají vzhled menu, pokud se horizontální navigace na šířku displeje již nevejde a je nutné jej udělat vertikální v podobě rozklikávacího menu.

Hlavní obsahová část je rozdělená na dva sloupce. Toho docílíme použitím gridovacího systému. Levý sloupec bude zabírat pouze 4/12 celkové šířky, respektive 4 pomyslné sloupce z 12 na středních až velkých zařízeních a celou šířku na zařízeních malých. Obalovému elementu tedy přiřadíme Bootstrap grid třídu `.col-md-4 col-sm-12`. Pravý obsahový sloupec bude tedy využívat zbylých 8 pomyslných sloupců na středních až velkých zařízeních a celou šířku na zařízeních malých. Tomuto elementu přiřadíme tedy Bootstrap grid třídu `.col-md-8 col-sm-12`. Díky tomuto systému docílíme zobrazení těchto sloupců vedle sebe na středních až velkých zařízeních, a naopak jejich seskupení pod sebe na zařízeních s malým zobrazovacím displejem.

S takto nemalým úsilím a s téměř žádným zásahem do kaskádových stylů výsledné stránky, jsme

---

<sup>32</sup><http://getbootstrap.com/components/#navbar>

schopni snadno a rychle vytvořit responzivní webovou stránku. Tato ukázka názorně demonstruje sílu tohoto frameworku a tedy rychlost a jednoduchost tvorby základních webových stránek. Výsledná podoba se zdrojovými kódy je součástí práce jako Příloha 2 na CD/DVD. Pro komplexnější webové stránky je již nutno na specifické elementy použít své kaskádové stylování.

## 6 WordPress v praxi

V této části přistupuji k praktickému použití redakčního systému WordPress. Najdete zde popis samotné instalace tohoto systému, podrobně si popíšeme tvorbu vlastních šablon a naučíme se psát vlastní externí balíčky, tzv. plugíny.

### 6.1 Instalace WordPressu

Před samotnou instalací je nutno podotknout, že WordPress má své jisté požadavky na parametry hostujícího serveru. WordPress od verze 3.2 do dnešní (verze 4.7.4) doporučuje tyto parametry pro server:

- PHP ve verzi 5.2.4 nebo větší
- MySQL ve verzi 5.0.15 nebo větší, popřípadě jakoukoliv verzi MariaDB
- Povolen modul `mod_rewrite`<sup>33</sup>
- Jako systémová platforma je doporučen Linux, ovšem podporovaný jsou samozřejmě i systémy Windows a IOS.
- Webový server Apache nebo NGINX, popřípadě je možno použít i jiný webový server s podporou výše zmíněného, nicméně WordPress u jiných serverů nezaručuje správnou funkčnost.

Podrobnější výpis požadavků na server pro nejnovější verzi WordPressu najdete vždy na jejich stránkách<sup>34</sup>. Mezi většinou větších poskytovatelů serverových či hostingových řešení je dnes tohle nastavení standardem, tudíž by neměla nastat žádná zpětná nekompatibilita.

Samotná instalace spočívá v několika málo krocích a její jednoduchost patří mezi hlavní devízu WordPressu. Kroky instalace jsou následující:

1. Stažení instalačního balíčku WordPressu: Instalační balíček nalezneme na oficiálních stránkách WordPressu v sekci Download<sup>35</sup>. Na těchto stránkách nalezneme vždy jeho aktuální verzi.
2. Rozbalení balíčku a nahrání na server: Stáhnutý instalační balíček je v podobě ZIP archívu. Tento archív stačí rozbalit a jeho obsah nahrát na vlastní hostingový server.
3. Nainstalování WordPressu: WordPress se instaluje skrze webové rozhraní. Po zadání URL adresy vašeho serveru, se zobrazí grafický průvodce instalací systému. V jednotlivých krocích jste vyzváni k zadání nepostradatelných údajů k samotnému chodu systému. Jsou jimi především přístupové údaje k databázovému úložišti, přesněji tedy jméno databáze, jméno uživatele, heslo, databázový hostitel. A dále pak základní informace o stránce, jako název stránky, uživatelské jméno a heslo pro vstup do administrace.
4. Přihlášení do administrace: V posledním kroku instalace jste vyzváni k prvnímu přihlášení do administrace systému WordPress pomocí přihlašovacích údajů, které byly nastaveny při instalaci. Tímto samotná instalace končí a WordPress je plně připraven k použití.

---

<sup>33</sup>[http://httpd.apache.org/docs/current/mod/mod\\_rewrite.html](http://httpd.apache.org/docs/current/mod/mod_rewrite.html)

<sup>34</sup>[http://codex.wordpress.org/Hosting\\_WordPress](http://codex.wordpress.org/Hosting_WordPress)

<sup>35</sup><https://wordpress.org/download/>

## 6.2 WordPress šablony

Pro změnu výsledného vzhledu stránky slouží šablony. Těchto šablon může být ve WordPressu nainstalováno hned několik, aktivní je však vždy pouze jedna. Jako výchozí šablona WordPressu již několik let slouží šablona s názvem Twenty (současná verze Twenty Sventeen)<sup>36</sup>, která s každou verzí systému WordPress prochází aktualizací a vždy dodržuje moderní trendy a standardy. Díky obrovské oblíbenosti WordPressu, lze na internetu najít tisíce různých šablon od těch úplně jednoduchých, které jsou zdarma, až po placené komerční šablony. V této práci však tvoříme šablonu vlastní.

### 6.2.1 Anatomie šablon

WordPress šablony se defaultně nacházejí v podadresářích adresáře `wp-content/themes/`. Podadresáře samotných šablon udržují všechny důležité soubory šablony, jako soubory s kaskádovými styly, soubory s jednotlivými skriptovacími šablonami, volitelné funkční soubory se změnami systému WordPress, doplňkové soubory (javascriptové knihovny, flash animace, a další) a obrázky šablony. Například, šablona s názvem `test` by se nacházela v adresáři `wp-content/themes/test/`. WordPress nepovoluje čísla v názvu šablony, v takovém případě je šabloně znemožněno zobrazení v seznamu dostupných témat.

WordPress šablony se typicky skládají ze třech hlavních typů souborů, vyjma obrázků a javascriptových souborů:

1. Šablona stylů s názvem `style.css`, která řídí samotnou prezentaci stránky (vizuální design a rozvržení) webových stránek.
2. Soubory šablon (neplést se samotnými balíčky šablon vzhledu) s příponou `.php`, které řídí to, jakým způsobem webová stránka generuje informace z WordPress databáze.
3. Volitelný soubor s dodatečnými funkcemi šablony, přesněji soubor `functions.php`, který jsme již nakousli v Kapitole 4.5.3 o architektuře WordPressu a který se stará o změnu chování samotného systému při generování informací z databáze.

Vytvoření vlastní šablony lze docílit dvěma způsoby, vytvoření tzv. šablony potomka, nebo vytvoření šablony od základů.

### 6.2.2 Šablona stylů

Kromě informací v podobě CSS stylů ohledně výsledného designu a rozložení stránek, tento soubor také poskytuje podrobné informace o samotném tématu v podobě komentáře, který se musí nacházet na úplném začátku tohoto souboru. Tento komentář s informacemi o šabloně v záhlaví souboru `style.css` je povinný. Bez těchto informací bude šabloně opět znemožněno zobrazení v seznamu dostupných témat. Taktéž žádné dvě šablony nemohou mít stejné informace ve svém záhlaví, což by vedlo k problémům v dialogovém okně administrace pro výběr motivu. Jako výchozí vzor pro tyto dodatečné informace o šabloně se doporučuje šablona základního WordPress tématu (viz úvod Kapitoly 6.2) v jeho nejnovější verzi. Současnou podobu můžete vidět na Výpisu kódu 1.

---

1 /\*

2 *Theme Name: Název šablony*

<sup>36</sup><https://wordpress.org/themes/twentyfifteen/>

```

3 Theme URI: URL šablony (pokud je možnost jejího stažení)
4 Author: Autor šablony
5 Author URI: URL autora
6 Description: Samotný popis šablony
7 Version: Verze šablony (kvůli případným aktualizacím šablony)
8 License: Licence (pokud šablona nějaké podléhá)
9 License URI: URL na licenci výše uvedené
10 Text Domain: Lokalizační soubor.
11 Tags: Jednotlivé tagy šablony (malými písmeny, odděleny čárkami)
12 */

```

---

Výpis 1: Vzor informací o šabloně v souboru style.css.

Ze samotného popisu jednotlivých položek, je vcelku jasné, co která znamená. Za zmínku stojí pouze atribut **Tag** a **Text Domain**. Pokud se nejedná o komerční šablonu na zakázku, nýbrž o šablonu volně dostupnou, WordPress doporučuje v atributu **Tag** používat běžně užívané výrazy. Toto dovolí uživatelům najít vaši šablonu podle filtru těchto běžně užívaných tagů. Kompletní výpis těchto zavedených tagů naleznete v dokumentaci<sup>37</sup>. Atribut **Text Domain** nese název lokalizačního souboru, který je načten, pokud je šablona uzpůsobena pro svou lokalizaci do jiných jazyků.

### 6.2.3 Soubor s funkcemi šablony

Šablona může obsahovat soubor dodatečných funkcí, které se nachází v adresáři šablony v souboru s názvem **functions.php**. Tento soubor funguje v podstatě jako dodatečný plugin, a pokud je v adresáři se šablonou přítomen, je automaticky načten při samotné inicializaci WordPressu. Doporučené použití tohoto souboru je zejména pro:

- Aktivování funkcí šablony jako je boční panel, více navigačních menu, formát výstupu příspěvku a další. Seznam všech dostupných funkcí, jejich popis a použití naleznete v dokumentaci pod záložkou **Theme Features**<sup>38</sup>.
- Definování funkcí, které lze použít ve všech souborech šablon naší šablony.
- Vytvoření nastavení naší šablony, které se zobrazí v administraci, dovolující uživateli měnit obsah nebo vzhled stránek bez nutnosti zasáhnout do zdrojového kódu. Jako příklad může posloužit patička webu, nesoucí kontaktní informace, které lze kdykoliv změnit v grafickém GUI administrace.

Nutno podotknout, že funkce nacházející se v souboru functions.php jsou dostupné jenom pro danou šablonu s tímto funkčním souborem. Funkce, které si přejeme globálně viditelné pro všechna témata se vytvářejí formou přídatného pluginu.

### 6.2.4 Soubory šablon

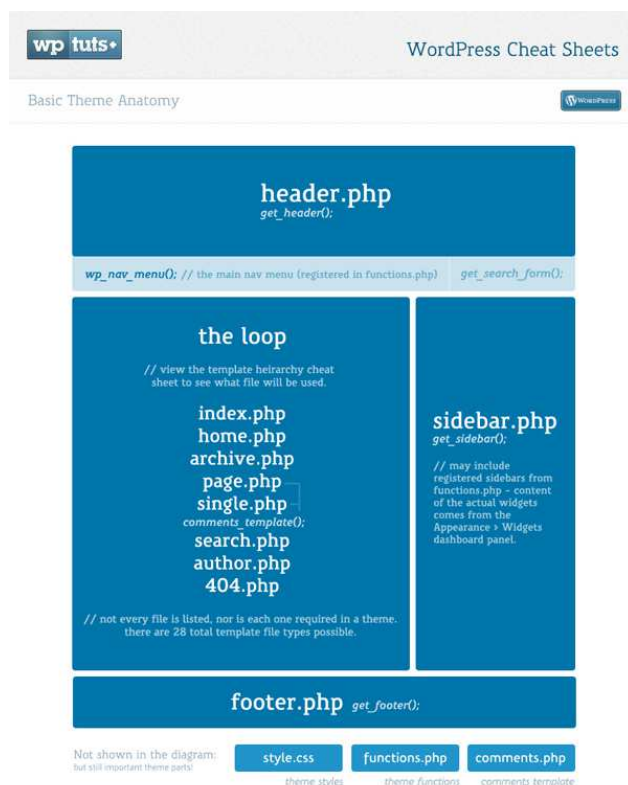
Soubory šablon jsou PHP zdrojové soubory používané ke generování stránek. Obsahují požadavky uživatele ve formě HTML kódu, formátovaného ve zvolené podobě. Zjednodušeně řečeno, jedná se o několik souborů s předem definovaným formátem, jejímž skládáním generujeme finální podobu

---

<sup>37</sup><https://make.wordpress.org/themes/handbook/review/required/theme-tags/>

<sup>38</sup>[https://codex.wordpress.org/Theme\\_Features](https://codex.wordpress.org/Theme_Features)

webové stránky postavené na WordPressu. WordPress umožňuje programátorovi definovat jednotlivé soubory šablon pro různé aspekty webové stránky. Není povinné, aby naše šablona obsahovala všechny tyto soubory. Vývojář může šablonu složit z jednoho samotného souboru *index.php*, který se bude starat o všechny výpisy a zobrazení stránek, avšak pro plnou funkčnost šablony a maximální přizpůsobení šablony, je vhodné definovat všechny tyto soubory. Jak se celá taková šablona skládá a kde jsou jednotlivé soubory šablon načítány, můžete vidět na obrázku 27. Kompletní hierarchie jednotlivých souborů šablony a jejich postupné volání, je nejlépe vidět v Příloze D.



Obrázek 27: Ukázka skládání šablony.

Zdroj: <http://code.tutsplus.com/articles/wordpress-cheat-sheets-theme-anatomy-model-wp-22579>

Zcela na začátku se WordPress „zeptá“ o jakou stránku se jedná a pro tuhle stránku hledá vhodnou šablonu, přičemž sestupuje celou hierarchií níže (na obrázku Přílohy D zleva doprava), dokud šablonu nenajde. V případě nalezení požadované šablony, WordPress soubor načte a dle jeho konfigurace vypíše obsah. Jak můžeme vidět, pokud není nalezena žádná šablona, WordPress sestoupí až na poslední šablonu *index.php*, která se o samotný výpis postará.

Seznam souborů šablon rozpoznatelných systémem WordPress a jejich krátký popis naleznete v seznamu níže. Kompletní popis hierarchie šablon naleznete v dokumentaci v sekci Theme Development<sup>39</sup>.

- **style.css** - Hlavní soubor kaskádových stylů. Musí být součástí každé šablony a musí obsahovat informace o šabloně ve svém záhlaví, viz Kapitola 6.2.2.

<sup>39</sup>[https://codex.wordpress.org/Theme\\_Development](https://codex.wordpress.org/Theme_Development)

- *index.php* - Hlavní šablona, do které se následně načítají ostatní definované šablony. Musí být přítomna v každé šabloně.
- *comments.php* - Šablona generující výpis komentářů, tedy jakým stylem je zobrazeno komentářové vlákno.
- *home.php* - Šablona hlavní stránky, která je ve výchozím stavu nastavena jako šablona pro zobrazení nejnovějších příspěvků. Pokud šablona neexistuje, WordPress použije šablonu *index.php*.
- *front-page.php* - Šablona hlavní stránky, která dle nastavení WordPressu zobrazí buď úvodní stránku jako statickou, tedy předá výpis šabloně *page.php* nebo na úvodní stránce zobrazí výpis nejnovějších příspěvků, tedy předá výpis šabloně *home.php*.
- *single.php* - Šablona detailu otevřeného příspěvku.
- *single-{typ-postu}.php* - Slouží pro zobrazení detailu, pokud příspěvek je typu tzv. Custom Post<sup>40</sup>, tedy příspěvku s odlišným formátem výpisu definovaného při tvorbě šablony. Velmi často použitelný u firemních prezentací na WordPressu, kdy je potřeba, aby každý článek nebo stránka byla vykreslena jiným stylem.
- *page.php* - Šablona statické stránky.
- *sidebar.php* - Šablona postranního panelu, může obsahovat námi zvolené výpisy, definované v tomto souboru.
- *category.php* - Šablona pro výpis hledání dle zvolené kategorie.
- *tag.php* - Šablona pro výpis hledání dle zvoleného tagu.
- *author.php* - Šablona pro výpis hledání dle zvoleného autora.
- *date.php* - Šablona pro výpis hledání dle zvoleného data či času.
- *archive.php* - Šablona, která defaultně zobrazuje výpis výsledků hledání pro výše zmíněné šablony. Pokud některá z výše uvedených šablon pro zobrazení výsledku hledání existuje, přepisuje tuto šablonu. V opačném případě se o výpis postará tato šablona.
- *search.php* - Stránka sloužící k výpisu dle výrazů zadaných v okně pro vyhledávání. Šablona hledá výrazy jak v nadpisech jednotlivých článků, tak i v samotných textech jednotlivých článků.
- *attachment.php* - Díky této šabloně můžeme upravovat výpis pro soubory, specifikované dle jeho typu.
- *404.php* - Pokud server odpovídá chybou 404, tedy v případě, že WordPress nemohl nalézt příspěvek nebo stránku na základě zadaných kritérií, zobrazuje se tato šablona.

---

<sup>40</sup>[https://codex.wordpress.org/Post\\_Types](https://codex.wordpress.org/Post_Types)

## 6.3 Tvorba šablony v praxi

Jelikož už chápeme základní strukturu šablon a jejich hierarchii, můžeme se pustit do popisu praktické části. Ukážeme si jak výše zmíněné informace prakticky použít a vytvoříme si vlastní WordPress šablonu. [3]

Všechny dosažené znalosti systému demonstrujeme na webovém portálu pro základní a mateřskou školu postaveném na WordPressu. Na šabloně si názorně předvedeme její kompletnost, úpravy nad rámec klasické struktury blogových layoutů a také si předvedeme tvorby jednoduchých WordPress přídatných modulů. Výsledná podoba webového portálu je součástí Přílohy E. Výsledná šablona a její zdrojové kódy je také součástí Přílohy 3 na CD/DVD. Digitální podoba webového portálu se nachází na adrese <http://hornane.cz/DOBES/wordpress/>.

### 6.3.1 Šablony webu

**Index.php** Hlavní šablona každého webu je bezpochyby soubor *index.php*. V našem případě šablona obsahuje pouze nezbytné minimum. Staráme se zde o načtení ostatních souborů, které se již starají o výpis samotný. Na začátku načítáme soubor s hlavičkou webu, na což slouží WordPress vestavěné funkce pro volání známým souborům šablon. Hlavičku načteme příkazem *get\_header()*. Následně na to načítáme do šablony boční postranní panel, který voláme obdobně jako hlavičku příkazem *get\_sidebar()*. Tyto funkce pouze vyhledají v adresáři šablony příslušné soubory, které musí být tedy pojmenovány dle výše zmíněného předpisu. Nyní následuje smyčka, starající se o vykreslení samotného obsahu. Nejprve se ujistíme, že v systému existují články, které se mají vypsát. Slouží nám pro to vestavěná funkce *have\_posts()*, která navrácí logické *true* nebo *false* v závislosti na tom, zda se v databázi nacházejí články k vykreslení. Pokud tomu tak je, následuje samotná smyčka vykreslení příspěvků. Tuto smyčku můžete vidět na výpisu zdrojového kódu 2.

---

```
1 while ( have_posts() ) {  
2     the_post();  
3     get_template_part( 'content', get_post_format() );  
4 }
```

---

Výpis 2: Vykreslovací smyčka hlavní strany.

První funkce *have\_posts()* s návratovou hodnotou *bool* říká, zda vykreslovací smyčka stále obsahuje příspěvky a následná funkce *the\_post()* se stará o iteraci indexu příspěvku ve smyčce. Následuje samotné vykreslení příspěvků. Pro tuto funkci voláme šablonu *content.php*, která se o tento výpis stará. Více o ní si řekneme níže. Po vypsání všech příspěvků voláme funkci pro zobrazení stránkování. Stránkování se zobrazí v případě, že počet článků přesahuje maximálně možný počet na jednu stránku dle hodnoty nastavené v administraci stránky. Stránkování se volá pomocí funkce *the\_posts\_pagination()*.

Pokud stránka neobsahuje žádné příspěvky k zobrazení (dle podmínky výše), pro výpis se volá jiná šablona, starající se o výpis prázdné stránky. V našem případě tedy voláme soubor *content-none.php*, který voláme obdobně jako stránku s příspěvky a to funkcí *get\_template\_part( 'content-none' )*.

Na samotném konci voláme patičku celé stránky, přesněji soubor *footer.php*. O načtení patičky se stará vestavěná funkce *get\_footer()*.



**Header.php** Soubor šablony header.php obsahuje kompletní hlavičku HTML stránky, ale také samotnou hlavičku webu, což je tedy především hlavička s logem a popisem stránky, okno pro vyhledávání a také hlavní menu.

Jedinými změnami hlavičky HTML stránky oproti statické verzi, je změna jazykové sady webu, která musí odpovídat jazykové sadě výchozí instalace WordPressu. Správnou jazykovou sadu získáme funkcí `bloginfo('charset')`, s návratovým typem *string*. Další změnou je připojení vlastního souboru s kaskádovými styly, který si WordPress při aktivaci šablony registruje. Soubor zavoláme stejnou funkcí jako jazykovou sadu, akorát z jiným parametrem, a to `bloginfo('stylesheet_url')`. Seznam všech dostupných parametrů této funkce je k nalezení v dokumentaci funkce `bloginfo()`<sup>41</sup>.

Následuje hlavička webové stránky, která spočívá ve vypsání popisu webu, který byl zadán v samotném nastavení webu. Popis webu získáme opět pomocí funkce `bloginfo()`, přesněji společně s parametrem `bloginfo('description')`. V neposlední řadě potřebujeme zobrazit okno pro vyhledávání, o které se stará šablona `searchform.php`, kterou můžeme načíst pomocí funkce `get_search_form()`.

Nakonec si zobrazíme hlavní menu stránky. Hlavní menu se vypisuje pomocí funkce `wp_nav_menu ( array $args = array() )`, kterou voláme tam, kde chceme zobrazit uspořádaný seznam s jednotlivými položkami menu. Seznam všech dostupných parametrů je k nalezení v dokumentaci<sup>42</sup>. V našem případě používáme tyto parametry:

1. menu - Označuje menu k zobrazení (v případě, že máme registrovaných více menu). Může být zadán jeho názvem, ID, nebo zkratkou.
2. depth - Určuje kolik úrovní hierarchie chceme zobrazit (výchozí 0 = všechny úrovně). Jelikož si přejeme následné podmenu zobrazit v bočním panelu, celou hierarchii vykreslovat nebudeme.
3. container - Určuje jakým HTML elementem chceme obalit celé menu (výchozí je „div“).
4. menu\_class - Třída obalového elementu „ul“, který obaluje celé menu (výchozí je „menu“). Jelikož chceme použít Bootstrap komponentu navigace, která počítá s třídou „nav“, je nutno výchozí třídu změnit.
5. walker - Název funkce tzv. walker třídy, která byla vytvořena speciálně kvůli Bootstrapu. Dovoluje procházet stromovou strukturu dat s položkami menu pro účely vykreslení HTML. Více o této speciální třídě se můžete dočíst v její referenci<sup>43</sup>.

**Footer.php** Tato šablona slouží k vykreslení patičky webu. V našem případě se v patičce vykreslují dodatečné informace jako krátký kontakt, odkaz na sociální sítě, právní informace a rychlá navigace. Všechny tyto položky jsou plně modifikovatelné z prostředí administrace, tudíž je uživateli umožněno jejich obsah měnit bez nutnosti zásahu do zdrojového kódu. Všechny tyto informace voláme pomocí funkce `get_option( $option, $default )`, o které si něco povíme v kapitole 6.3.2, kde si také ukážeme její použití. V poslední řadě ještě připojíme vlastní javascriptový soubor. Pro nalezení cesty nám poslouží funkce `get_template_directory_uri()`, která vrátí URI adresáře s naší šablonou.

<sup>41</sup><https://developer.wordpress.org/reference/functions/bloginfo/>

<sup>42</sup>[https://developer.wordpress.org/reference/functions/wp\\_nav\\_menu/](https://developer.wordpress.org/reference/functions/wp_nav_menu/)

<sup>43</sup>[https://codex.wordpress.org/Class\\_Reference/Walker](https://codex.wordpress.org/Class_Reference/Walker)

**Content.php** Jedná se o vlastní šablonu, tzn. není to jedna ze základních rozpoznatelných šablon WordPressu. Šablona se stará o výpis obsahu na hlavní stránce. Každý příspěvek má svou hlavičku, kde najdeme titulek příspěvku a drobné meta informace o příspěvku a své tělo, které obsahuje samotný text příspěvku, popřípadě náhledový obrázek příspěvku. Titulek příspěvku vypíšeme vestavěnou funkcí `the_title()`. Dále voláme funkci `article_footer()`, což je jedna z našich vlastně definovaných funkcí, kterou si více popíšeme v Kapitole. Zjednodušeně se funkce stará o přizpůsobení výpisu meta informací dle našich kritérií. Jelikož WordPress poskytuje svůj standardní formát výpis, tak pro jeho změnu musíme udělat tento malý zásah do systému prostřednictvím souboru `functions.php`. Následuje volání samotného textu příspěvku, který také voláme pomocí své vlastní funkce `wp_trim_words_retain_formatting( $obsah, pocet_znaku )`, kterou najdete v souboru `functions.php`. WordPress nabízí vestavěnou funkci pro výpis obsahu a to voláním `the_content()`. Bohužel tato funkce byla pro naše potřeby nedostatečná. Cílem bylo ořezat text každého příspěvku na hlavní stránce na určitý počet znaků. Bohužel funkce `the_content()` vrací text příspěvku, který je obalen několika HTML divy, které obsahují obsáhlé popisy WordPress tříd, které se do samotného textu k ořezání také počítají. Cílem naší funkce bylo tedy odstranit tyto nepotřebné části ze samotného textu, který bude poslán k ořezání.

**Page.php** Šablona, kterou WordPress volá, pokud se jedná o statickou stránku. Šablona jako taková pouze volá vlastní šablonu `page.php`, která se o samotný výpis stará. Jelikož při požadavku na statickou stránku se nevolá šablona `index.php`, ale přímo tato šablona (viz hierarchie volání šablon), nesmíme zapomenou načíst hlavičku, postranní panel a patičku webu, jak tomu bylo u šablony `index.php`. Šablona pouze obsahuje vykreslovací smyčku, která pokud má co vykreslit, předává vykreslení šabloně `content-page.php`. Tato šablona je také velmi jednoduchá a vypisuje titulek stránky vestavěnou funkcí `the_title()` a samotný obsah stránky vestavěnou funkcí `the_content()`.

**Single.php** Jedná se o šablonu, která je volána pokud je vyžadován detail jednoho příspěvku. Tato šablona je opět volána namísto úvodní šablony `index.php`, nesmíme tedy zapomenout opět načíst hlavičku a patičku webu. Postranní panel zde nenačítáme, jelikož chceme příspěvek roztažený na plnou šíři stránky. Samotný výpis příspěvku předává šabloně `content-single.php`. Tato šablona opět vypisuje pouze titulek příspěvku vestavěnou funkcí `the_title()` a jeho obsah vestavěnou funkcí `the_content()`.

**Sidebar.php** Jedná se o šablonu s postranním panelem, která je poněkud specifická. Postranní panel bude zobrazovat podmenu hlavního menu. Hlavičku postranního panelu tvoří nadpis rodičovského odkazu hlavního menu, který, vypíšeme pomocí vestavěné funkce `get_the_title($post->post_parent)`. Kde proměnná `$post` obsahuje všechna meta data současného prvku a můžeme ji volat globálně. V našem případě tedy získáme ID svého rodiče, dle kterého vypíšeme jeho název. Poté následuje výpis samotných položek podmenu funkcí `wp_list_pages ( array/string $args = ' ' )`.

**Další šablony** Jelikož další šablony již neobsahují žádné nové funkce, můžeme je shrnout do jednoho bodu. Máme zde šablony zobrazující výpis příspěvků dle zadaných kritérií, jako `archive.php`, `category.php`, `comments.php` a `search.php`. Všechny tyto šablony svůj výpis předávají šabloně `content.php`. Tuto šablonu volám funkcí `get_template_part( 'content', get_post_format() )`, kde funkce `get_post_format()` vrací požadovaný formát výpisu dle toho, jaké volání WordPress provedl.

### 6.3.2 Soubor funkcí

Stěžejní soubor je bezpochyby soubor *functions.php*, které obsahuje všechny vlastní funkce, volané v ostatních šablonách, ale také dodatečné změny chování systému WordPress jako takového. Změn jsme provedli hned několik.

**Odlišný výpis patičky příspěvku** Nalezneme zde vlastní funkci, která se stará o modifikovaný výpis informací o příspěvku. Defaultně WordPress u každého příspěvku zobrazí pouze autora příspěvku. Pomocí vlastní funkce si můžeme tento výpis modifikovat a vypsat tedy více informací. V tomto konkrétním případě tedy autora článku, datum přidání, kategorii příspěvku a rychlý odkaz pro úpravu příspěvku. je k vidění v Příloze 3 na CD/DVD v souboru *functions.php*, řádek číslo 17.

**Více hlavních nabídek** WordPress primárně používá pouze jedno hlavní menu, což je pro naše potřeby nedostatečné, jelikož jsme chtěli oddělit vrchní hlavní menu a podmenu v postranním panelu. WordPress tedy dovoluje registraci vlastních menu pomocí funkce *register\_nav\_menus( array() )*. Parametr funkce je pole, které obsahuje dvojici identifikátoru menu a samotného názvu menu, tak jak se bude zobrazovat v administraci. Identifikátor menu slouží k specifikaci, které menu chceme vykreslit, viz Kapitola 6.3.1 v části o vykreslení menu - parametr „menu“. Vytvoření dvou odlišných menu (horizontální hlavní menu a vertikální submenu) je k vidění na výpisu zdrojového kódu 3.

---

```
1 register_nav_menus( array(
2     'primary' => __( 'Primary_Navigation', 'BacPrac' ),
3     'secondary' => __( 'Secondary_Navigation', 'BacPrac' )
4 ));
```

---

Výpis 3: Registrace více navigací.

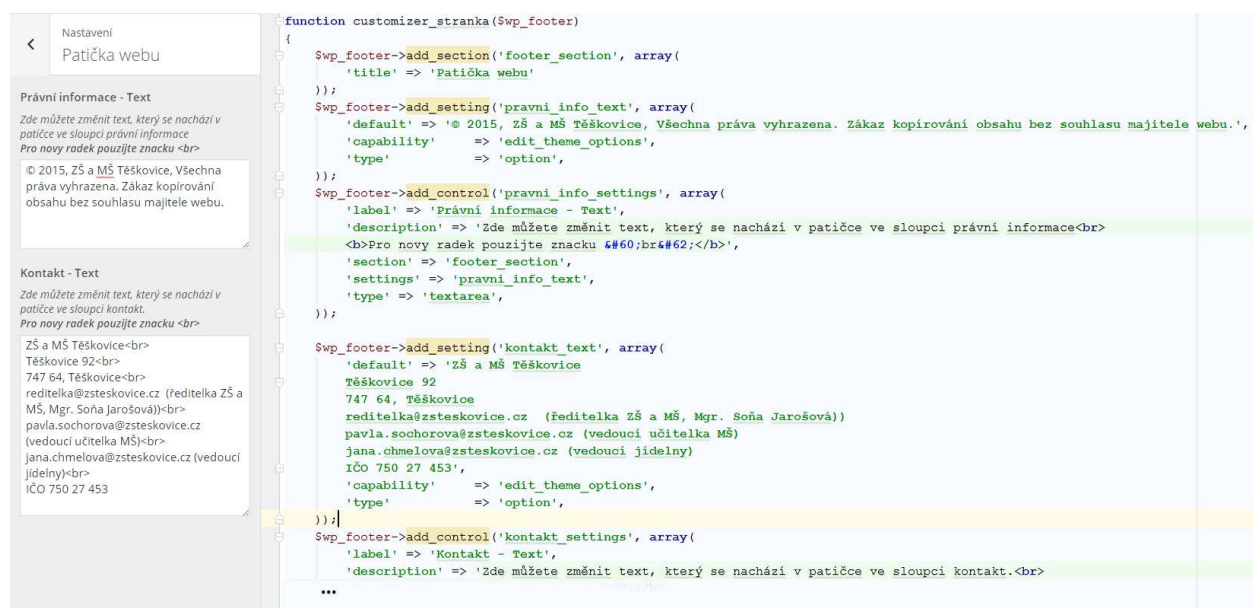
**Náhledový obrázek** Jelikož bylo požadavkem, aby jednotlivé příspěvky měly svůj náhledový obrázek, musíme přidat podporu této funkcionality. WordPress má v základu tuto funkci neaktivní, je nutné tedy přidat do šablony podporu této funkce. To provedeme vestavěnou funkcí *add\_theme\_support( 'post-thumbnails' )*. Všechny volitelně dodatečné funkce nalezneme v dokumentaci<sup>44</sup>.

**Uživatelsky modifikovatelná patička** WordPress nám umožňuje dynamicky měnit podobu šablony, které nejsou nijak dostupné prostřednictvím administrace. V našem případě požadujeme úpravu informací přímo z prostředí administrace bez nutnosti zásahu do zdrojového kódu. WordPress tedy umožňuje vytváření specifických nastavení šablony, která je poté dostupná z administrace (Administrace -> Vzhled -> Přizpůsobení). Jedna definice konkrétního bloku nastavení se tváří jako funkce, která nese několik informací. Funkcí *add\_section()* se specifikuje daný blok s názvem, který se bude zobrazovat jako nová nabídka v přizpůsobení šablony. Funkcí *add\_setting()* se dále specifikuje nastavení bloku pro jeden údaj nastavení a funkcí *add\_control()* se poté nastaví daný input<sup>45</sup>. Zaregistrování tohoto nového nastavení se provede pomocí funkce *add\_action( 'customize\_register', 'nazev\_funkce\_s\_definici' )*. Ukázku zdrojového kódu a jeho výsledek v prostředí administrace můžete vidět na obrázku 28. Tyto uložené informace můžeme následně v samotné šabloně volat funkcí

<sup>44</sup>[https://developer.wordpress.org/reference/functions/add\\_theme\\_support/](https://developer.wordpress.org/reference/functions/add_theme_support/)

<sup>45</sup><http://jecas.cz/input>

`get_option()`. Kompletní zdrojový je k vidění v Příloze 3 na CD/DVD v souboru *functions.php*, řádek číslo 40.



Obrázek 28: Ukázka vytvoření vlastního nastavení webu.

**Úprava uživatelských rolí** WordPressu v základu nabízí správu registrovaných uživatelů a také přidělení určitých rolí. WordPress nabízí dvě možnosti registrace uživatele. Manuální, kdy administrátor musí ručně nového uživatele vytvořit a jemu pak budou přihlašovací údaje automaticky zaslány na email, a registraci přes registrační formulář. Úroveň respektive role uživatele, definuje oprávnění uživatele v administraci WordPressu nebo na jeho podstránkách. V základním nastavení disponuje WordPress pěti úrovněmi uživatelských rolí:

1. **Administrátor** - Má neomezený přístup ke všem nastavením WordPressu. Má dovoleno dělat vše, co je ve WordPressu možné, instalovat Šablony, pluginy, přidávat a odebírat uživatele, měnit jim oprávnění, prostě pracovat kompletně se všemi dostupnými nástroji.
2. **Šéfredaktor** - Nemá přístup k pluginům, šablonám, nebo widgetům, ale můžete spravovat kategorie, média, články, odkazy, komentáře, zkrátka kompletní obsahovou část webu. Nemá přístup k žádnému nastavení webu.
3. **Redaktor** - Může vytvářet, publikovat a mazat své vlastní články, nahrávat média a může mazat, to co dříve nahrál. Také může moderovat komentáře u jeho postů. Nemá, na rozdíl od Šéfredaktora, přístup k obsahu ostatních uživatelů.
4. **Spolupracovník** - Spolupracovník má k dispozici vytváření, úpravu a mazání vlastních, doposud neuveřejněných článků. Každý článek, který napíše, nebude publikován a musí jej schválit Šéfredaktor. Důležité je, že spolupracovník nemá přístup do médií a pokud chce do článku vložit nějaký obrázek, musí požádat o pomoc Šéfredaktora.

5. Návštěvník - Návštěvník je základní oprávnění nového uživatele po registraci do WordPressu. Návštěvník může číst články, vkládat komentáře a upravovat svůj profil. Nikam jinam nemá přístup.

Naším úkolem bylo nepotřebné role odstranit a vytvořit role vlastní pro naše záměry. Stávající role WordPressu jsou přenášeny jako pole globální proměnné `$roles[]`, pro jejich odstranění stačí tyto proměnné smazat. Například pro smazání role Redaktor použijeme příkaz `unset($roles['editor'])`. Pro vytvoření vlastních uživatelských rolí slouží funkce `add_role()`, jejichž kompletní popis najdete v dokumentaci<sup>46</sup>. Zkrácený výpis zdrojového kódu pro vytvoření vlastní uživatelské role můžeme vidět na výpisu kódu 4, kompletní zdrojový kód je k vidění v Příloze 3 na CD/DVD v souboru `functions.php`, řádek číslo 145.

---

```
1 $result = add_role( 'ucitel1', __( 'Učitel_-_nižší_oprávnění' ),
2     array(
3         'activate_plugins' => false,
4         'create_posts' => true,
5         'delete_others_pages' => false,
6         'delete_others_posts' => false,
7         'delete_pages' => false,
8         'delete_posts' => true,
9         'delete_private_pages' => false,
10        'delete_private_posts' => false,
11        .....
12    ));
```

---

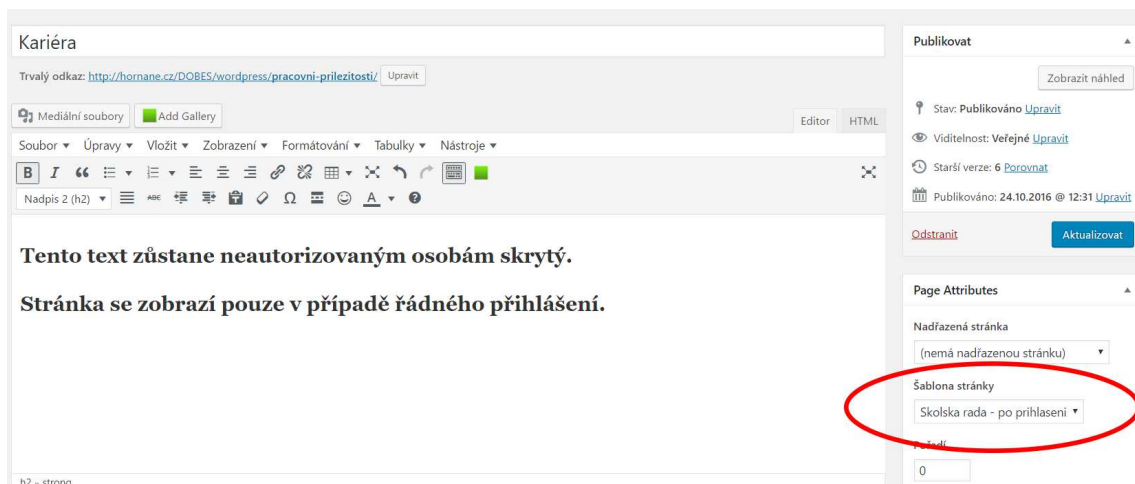
Výpis 4: Zkrácený výpis vytvoření vlastní uživatelské role.

### 6.3.3 Šablony stránky

Můžeme samozřejmě také odlišit, jak se bude která vytvořená stránka chovat. Slouží k tomu tzv. šablony stránky neboli page templates. Její název začíná klíčovým slovem `page_` jako klasická základní šablona `page.php` po podtržítku následuje vlastní název dané šablony. Klíčovým aspektem je hlavička tohoto souboru s šablonou, dle které systém WordPress pozná, že se jedná právě o šablonu stránky. Na samotném první řádku souboru musí být klíčový komentář `<?php /* Template Name: NAZEV_SABLONY */ ?>`, po kterém již následuje samotná struktura výpisu dané stránky. V našem konkrétním případě jsme vytvořili šablonu Školská rada, soubor `page_rada.php`, který poskytuje svůj obsah pouze řádně autorizovaným uživatelům, dle uživatelských rolí. Uvnitř souboru používáme svou funkci ze souboru `functions.php` pro zjištění autorizace, na jejímž základě stránku zobrazíme nebo zobrazíme pouze informační zprávu o nedostatečném oprávnění ke shlédnutí stránky. Na obrázku můžete vidět, jak danou šablonu jednoduše přiřadíme ke stránce, jejichž obsah chceme zobrazit pouze autorizovaným osobám.

---

<sup>46</sup>[https://codex.wordpress.org/Function\\_Reference/add\\_role](https://codex.wordpress.org/Function_Reference/add_role)



Obrázek 29: Ukázka přiřazení šablony ke stránce

## 6.4 Tvorba pluginu

### 6.4.1 Co je to WordPress plugin

Plugin, neboli česky zásuvný modul, je malý program napsaný ve stejném jazyce (PHP), který pohání samotný systém WordPress. Jedná se o opakovatelně použitelný kód, neboli funkčnost systému, který můžeme přenášet s projektu na projekt. Pluginy přinášejí pro náš systém nové rozšíření a možnosti chování WordPress backendu i frontendu. Pokud je plugin k dispozici, je volán na začátku každé specifické operace systému. Například tedy, před samotným zobrazením příspěvku WordPress zkontroluje, zda není nainstalován plugin, starající se o specifické vykreslení dané operace. Pokud takový plugin aktivní je, převezme vykreslení příspěvku na sebe. Díky velké komunitě WordPressu je dnes k nalezení přes 50 000 různorodých pluginů volně ke stažení a další tisíce komerčních modulů. Pokud i přesto vhodný modul nenalezneme, jsme jako programátor nuceni vytvořit si vlastní plugin.

### 6.4.2 Hlavička pluginu

První krok při tvorbě pluginu je vytvoření nového PHP souboru pro zápis funkcionality modulu. Název tohoto souboru by měl být unikátní, jelikož všechny WordPress pluginy se nacházejí ve stejném adresáři a to `./wp-content/plugins`. Pokud by název pluginu byl až příliš obecný, je zde značné riziko, že náš plugin by kolidoval s jiným pluginem pod totožným názvem a riskujeme tedy, že systém WordPress náš plugin nezobrazí ve svém seznamu. Povinná složka pluginu je validní hlavička souboru. Tato hlavička musí být definována na samotném začátku hlavního PHP souboru. Tuto hlavičku již nemusí obsahovat další dodatečné soubory pluginu. Tato hlavička sděluje systému, že tento PHP soubor je ve skutečnosti legitimní zásuvný model WordPressu a systém s ním má také tak nakládat. Na výpisu kódu 5 můžete vidět, jak taková předepsaná hlavička má vypadat. Povinný parametr hlavičky je pouze Plugin Name. Ostatní informace jsou pouze volitelné, avšak dle předepsaného vzoru velmi doporučené. Tyto dodatečné informace jsou poté interpretovány sekci pro správu pluginů, jako můžete vidět na obrázku, kde je dobře vidět, proč jsou i tyto ostatní informace důležité.

---

```
1 /*
2 Plugin Name: Facebook Meta Tagy
3 Plugin URI: http://vsb.cz
4 Description: Přidání Facebook meta tagů do hlavičky webu.
5 Version: 1.0
6 Author: Vaclav Dobes
7 */
```

---

Výpis 5: Vzorová hlavička pluginu.

### 6.4.3 WordPress Hooks

Plugins provádí své chování na základě tzv. WordPress Hooks, dále jen háčky. Zjednodušeně řečeno, jedná se o prázdné místo ve WordPress šabloně, na které při správném volání může plugin vkládat svůj výpis. Základní WordPress háček se nazývá `wp_head()`. Každé nové téma je povinno mít ve své hlavičce tuhle funkci. Seznam všech WordPress háčku můžeme najít v dokumentaci<sup>47</sup>. Najdeme zde velký výčet všech háčků v samotné struktuře systému WordPress, tak i volitelné háčky, které můžeme přidat do naší vlastní šablony. Chceme-li tedy vypsát něco do hlavičky šablony, využijeme právě tohoto háčku. K tomuto účelu slouží funkce `add_action()`, nesoucí dva vstupní parametry: první z nich je název háčku a druhý funkce, která se má na tomto místě vykonat. Pro demonstraci háčků si vytvoříme jednoduchý plugin pro přidání facebook meta tagů<sup>48,49</sup> do hlavičky příspěvku. V podstatě se jedná o informace pro Facebook API při sdílení našich příspěvků na sociální síti, které udávají, jak má takový sdílený článek vypadat. Tedy, co je nadpis, článek, náhledový obrázek, atd. Vytvořenou funkci můžete vidět na obrázku 30 a kompletní plugin je také součástí Přílohy 4.1 na CD/DVD. V podstatě jsme zavolali háček `wp_head()`, na jehož místo vypsala funkce `moje_tagy()` své informace. Použili jsme vestavěnou WordPress funkci `is_single()`, která navrací návratovou BOOL hodnotu dle toho, zda-li se nacházíme na stránce s rozbaleným příspěvkem. Nejprve jsme tedy vypsali základní tagy dle template tags<sup>50</sup>. Poté jsme zkontrolovali, zda-li příspěvek obsahuje náhledový obrázek. Pokud ano, je přidán i meta tag pro náhledový obrázek u sdíleného článku Facebookem.

Další dva důležité háčky, které je potřeba zmínit jsou tzv. Activation / Deactivation Hooks. Pokud chceme využít některé důležité funkce při samotném aktivování pluginu, můžeme použít právě tyto háčky. První z nich je funkce `register_activation_hook`. Tato funkce se vykoná, když je váš plugin v systému aktivován. Funkce nese dva vstupní parametry: cestu k hlavnímu PHP souboru pluginu a název funkce, která se má vykonat při aktivaci pluginu. Tohle je velmi nápomocné, chceme-li vytvořit defaultní nastavení pluginu a vytvořit tak vlastní databázovou tabulku k uložení dat modulu. Druhá z nich je funkce, která se automaticky spustí, pokud je plugin deaktivován. Jedná se o funkci `register_deactivation_hook`. Tato funkce je vykonána, pokud je váš plugin v systému deaktivován. Funkce nese totožné dva vstupní parametry jako výše zmiňovaná funkce `register_activation_hook`. [4]

---

<sup>47</sup>[https://codex.wordpress.org/Plugin\\_API/Action\\_Reference](https://codex.wordpress.org/Plugin_API/Action_Reference)

<sup>48</sup><https://developers.facebook.com/docs/sharing/opengraph>

<sup>49</sup><https://developers.facebook.com/docs/sharing/best-practices>

<sup>50</sup>[https://codex.wordpress.org/Template\\_Tags](https://codex.wordpress.org/Template_Tags)

```

1 add_action( 'wp_head', 'moje_tagy' );
2 function moje_tagy() {
3     if( is_single() ) {
4         ?>
5         <meta property="og:title" content="<?php the_title() ?>" />
6         <meta property="og:site_name" content="<?php bloginfo( 'name' ) ?>" />
7         <meta property="og:url" content="<?php the_permalink() ?>" />
8         <meta property="og:description" content="<?php the_excerpt() ?>" />
9         <meta property="og:type" content="article" />
10
11         <?php
12             if ( has_post_thumbnail() ) :
13                 $image = wp_get_attachment_image_src( get_post_thumbnail_id(), 'large' );
14             ?>
15             <meta property="og:image" content="<?php echo $image[0]; ?>" />
16         <?php endif; ?>
17
18     <?php
19     }
20 }

```

Obrázek 30: Demonstrace WordPress háčků.

#### 6.4.4 WordPress akční filtry

Akční filtry jsou velmi podobné akčním háčkům, avšak můžou modifikovat část před použitím namísto implementování nových akcí. Nejlepší možný příklad je chybová zpráva, která se objeví při špatném pokusu o přihlášení do WordPress administrace. Akční filtry nám umožňují tuto zprávu generovanou WordPressem pozměnit. Pro demonstraci akčních filtrů si tedy vytvoříme velmi jednoduchý plugin, který tuto chybovou zprávu pozmění. K tomuto účelu slouží funkce *add\_filter()*, nesoucí dva vstupní parametry: prvním předáváme funkci data k modifikaci, druhým je název funkce, která data pozmění. Názorný příklad můžeme vidět na obrázku 31. Na dalším obrázku 32 můžeme poté vidět rozdíl se zapnutým pluginem (vpravo) a vypnutým pluginem (vlevo).

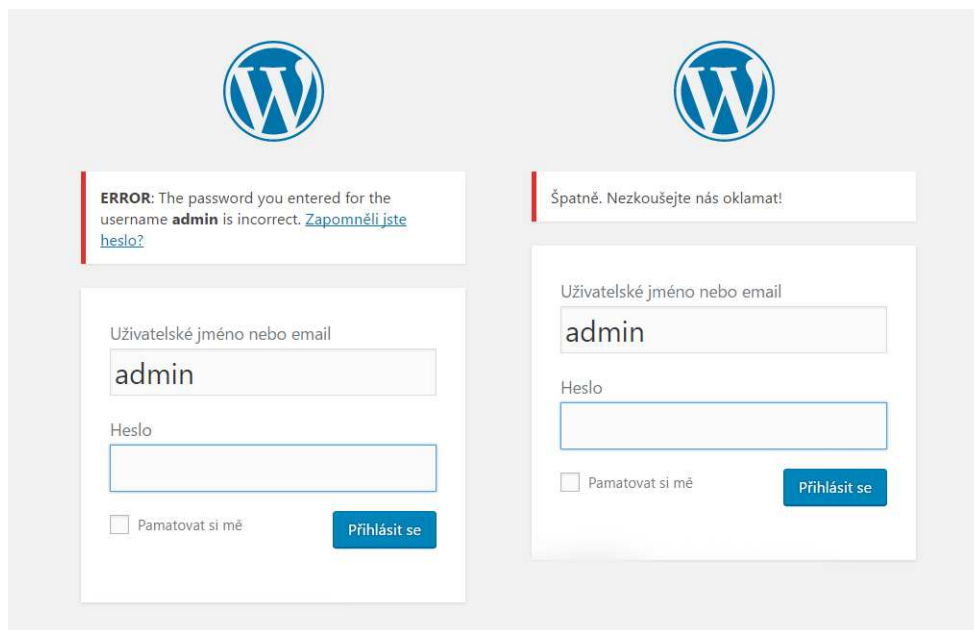
```

1 <?php
2 add_filter('login_errors', 'login_error_message');
3
4 function login_error_message( $error ){
5     $error = "Špatně. Nezkoušejte nás oklamat!";
6     return $error;
7 }

```

Obrázek 31: Demonstrace WordPress filtrů





Obrázek 32: Výsledek použití pluginu. Vypnutý (vlevo), zapnutý (vpravo).

#### 6.4.5 Nastavení pluginu z prostředí administrace

U některých pluginů můžeme vyžadovat jeho uživatelské nastavení přímo z prostředí administrace. V názorném příkladu se vrátíme k pluginu s podkapitoly 6.4.4 pro změnu chybové hlášky při přihlášení a nabídneme uživateli jednoduchou změnu chybové hlášky z prostředí administrace. V první řadě vytvoříme novou položku postranního menu WordPress administrace, která bude obsahovat uživatelské rozhraní pro požadované nastavení. Využíváme přitom háčku se vstupním parametrem `admin_menu`<sup>51</sup>, čímž docílíme, že požadovaná funkce druhého vstupního parametru se vykreslí právě v postranním menu administrace. Dále použijeme funkci `add_menu_page()`<sup>52</sup>, která má tyto vstupní parametry:

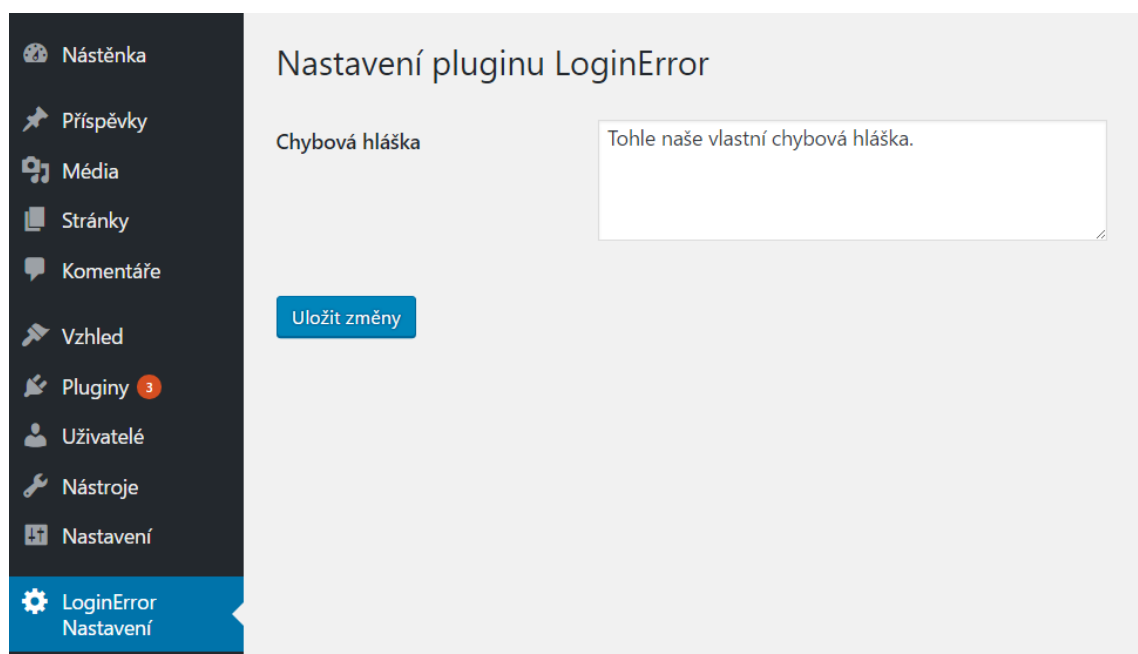
1. Page title – slouží jako titulek celého menu (panel v prohlížeči)
2. Menu title – slouží jako název položky menu v administraci
3. Capability – úroveň uživatele, který může k nastavení pluginu přistupovat
4. Menu slug – jedná se o parametr URL
5. Function – název funkce, která se bude starat o výpis uživatelského rozhraní
6. Icon – ikonka položky menu, může být zadána jako přímé URL na obrázek nebo volací znak vnořeným ikon Dashicons

Předtím nežli vytvoříme samotné uživatelské rozhraní, dáme WordPressu vědět jaké položky budeme potřebovat, přesněji řečeno, WordPress si zaregistruje do databázové tabulky pluginu požadované

<sup>51</sup>[https://codex.wordpress.org/Plugin\\_API/Action\\_Reference/admin\\_menu](https://codex.wordpress.org/Plugin_API/Action_Reference/admin_menu)

<sup>52</sup>[https://developer.wordpress.org/reference/functions/add\\_menu\\_page/](https://developer.wordpress.org/reference/functions/add_menu_page/)

stringové položky. V našem případě tedy potřebujeme uchovat zadanou chybovou hlášku. Použijeme tedy opět volací háček s parametrem `admin_init`<sup>53</sup>, ve kterém pomocí funkce `register_setting()`<sup>54</sup> zaregistrujeme naši novou položku, kterou plánujeme lokálně uložit. Následně můžeme vytvořit klasický HTML formulář, který bude sloužit jako uživatelské rozhraní. Uloženou hodnotu můžeme poté v pluginu volat funkcí `get_option()`<sup>55</sup>, jejichž první parametr je název stringu zadaný výše. Výsledná podoba pluginu je k dispozici jako Příloha 4.2 na CD/DVD. Podobu nové položky menu v administraci a její uživatelské rozhraní můžeme vidět na obrázku 33.



Obrázek 33: Nastavení pluginu z prostředí administrace

#### 6.4.6 Komplexnější pluginy

Pomocí výše zmíněných háčků a filtrů je možno poté skládat více komplexnější pluginy a rozšíření. Tvorba obsáhlejších pluginů se tedy bez pochopení výše zmíněného neobejde. Jako vzorový příklad rozsáhlejšího rozšíření systému WordPress poslouží plugin pro obarvení zdrojového kódu, který je k dispozici jako Příloha 4.3 na CD/DVD. Na okomentovaném zdrojovém kódu je názorně ukázáno, jak lze vytvořit rozsáhlejší rozšíření WordPressu. K tvorbě pluginu pomohla knihovna GeSHi<sup>56</sup>. Jedná se o volně dostupnou knihovnu pro zvýraznění syntaxe zdrojového kódu. Knihovna funguje na principu vstupního řetězce zdrojového kódu, který následně navrátí s přidánými inline CSS styly pro zvýraznění dané syntaxe. Nabízí také podporu číslování řádků a možnost výběru barevného schématu zvýraznění. Po aktivaci ukázkového pluginu najdeme ve vestavěném HTML editoru WordPressu nové tlačítko pro obalení zdrojového kódu do správného formátu, díky kterému se knihovna GeSHi postará o frontendové zvýraznění této syntaxe. Ukázkový plugin podporuje nativně pouze jazyk C++.

<sup>53</sup>[https://codex.wordpress.org/Plugin\\_API/Action\\_Reference/admin\\_init](https://codex.wordpress.org/Plugin_API/Action_Reference/admin_init)

<sup>54</sup>[https://codex.wordpress.org/Function\\_Reference/register\\_setting](https://codex.wordpress.org/Function_Reference/register_setting)

<sup>55</sup>[https://developer.wordpress.org/reference/functions/get\\_option/](https://developer.wordpress.org/reference/functions/get_option/)

<sup>56</sup><http://qbnz.com/highlighter/>

## Závěr

V práci jsme se zaměřili na statistiky použití moderních webových technologií, které jasně ukázaly, jakým směrem moderní web směřuje. Čtenáři jsme poskytli přehled o nových moderních trendech při návrhu responzivních webových stránek, ukázali nové webové technologie a nastínili jejich praktické využití. Podívali jsme se také na framework pro responzivní web a to Bootstrap. Ukázali jsme si, kde framework získat a jak jej začít používat. Rozebrali jsem jeho výsadní vlastnosti a nejpoužívanější komponenty. Na příkladu jsme si ukázali, jak rychle a jednoduše vytvořit webovou stránku s pomocí toho frameworku, s minimálním zásahem do stylovacích souborů. Dále jsme se zaměřili na redakční systém WordPress a čtenáři tak poskytli bližší pohled do nitra tohoto systému a ukázali jsme si, jak daný systém funguje a jak začít s tvorbou šablony či pluginu pro tento redakční systém.

Jelikož návodů v české jazyce pro tvorbu webových stránek pomocí těchto dvou mocných nástrojů je pramálo, poskytli jsme tak čtenáři názorný návod, jak s těmito nástroji začít pracovat. WordPress samotný disponuje obrovskou dokumentací, která začátečníkům nedává příliš šancí. Proto jsme čtenáři poskytli výčet nejdůležitějších vlastností a funkcí tohoto systému pro snadnější začátky s tímto mocným nástrojem.

Budoucnost autor vidí v elegantních jednostránkových prezentacích, které se velmi rozmáhají. V rámci budoucí tvorby se autor plánuje zaměřit a hlouběji proniknout do tajů a možností CSS koprocesorů LESS/SASS a poznat sílu těchto jazyků. Jelikož dnešní web pomalu opět sklouzává k elegantním animačním prvkům, bookům, či parallax efektům. Autor má v plánu se také zaměřit na nejpoužívanější JavaScriptovou knihu JQuery, která je také nedílnou součástí dnešních webových standardů. Tato bakalářská práce také pomohla zrodit několik dalších webových stránek na kombinaci WordPress a Bootstrap právě s podporou knihovny JQuery.

## Literatura

- [1] BROWN, Tiffany B., Kerry BUTTERS a Sandeep PANDA. HTML5 okamžitě: [ovládněte HTML5 za víkend]. Brno: Computer Press, 2014. ISBN 9788025142967.
- [2] GASSTON, Peter. CSS3. Přeložil Ondřej BAŠE. Brno: Computer Press, 2016. ISBN 9788025146415.
- [3] WILSON, Lisa Sabin. Wordpress all-in-one for dummies, 3rd ed. 3rd edition. ISBN 978-1119327776.
- [4] Getting Started with WordPress Plugin Development: The Ultimate Guide. Wpmudev [online]. 2017 [cit. 2017-04-20]. Dostupné z: <https://premium.wpmudev.org/blog/wordpress-plugin-development-guide/>

## Seznam příloh

Příloha A - Podrobná architektura WordPressu

Příloha B - Databázový model WordPressu

Příloha C - Bootstrap komponenty

Příloha D - Hierarchie WordPress šablon

Příloha E - Výsledná podoba webového portálu

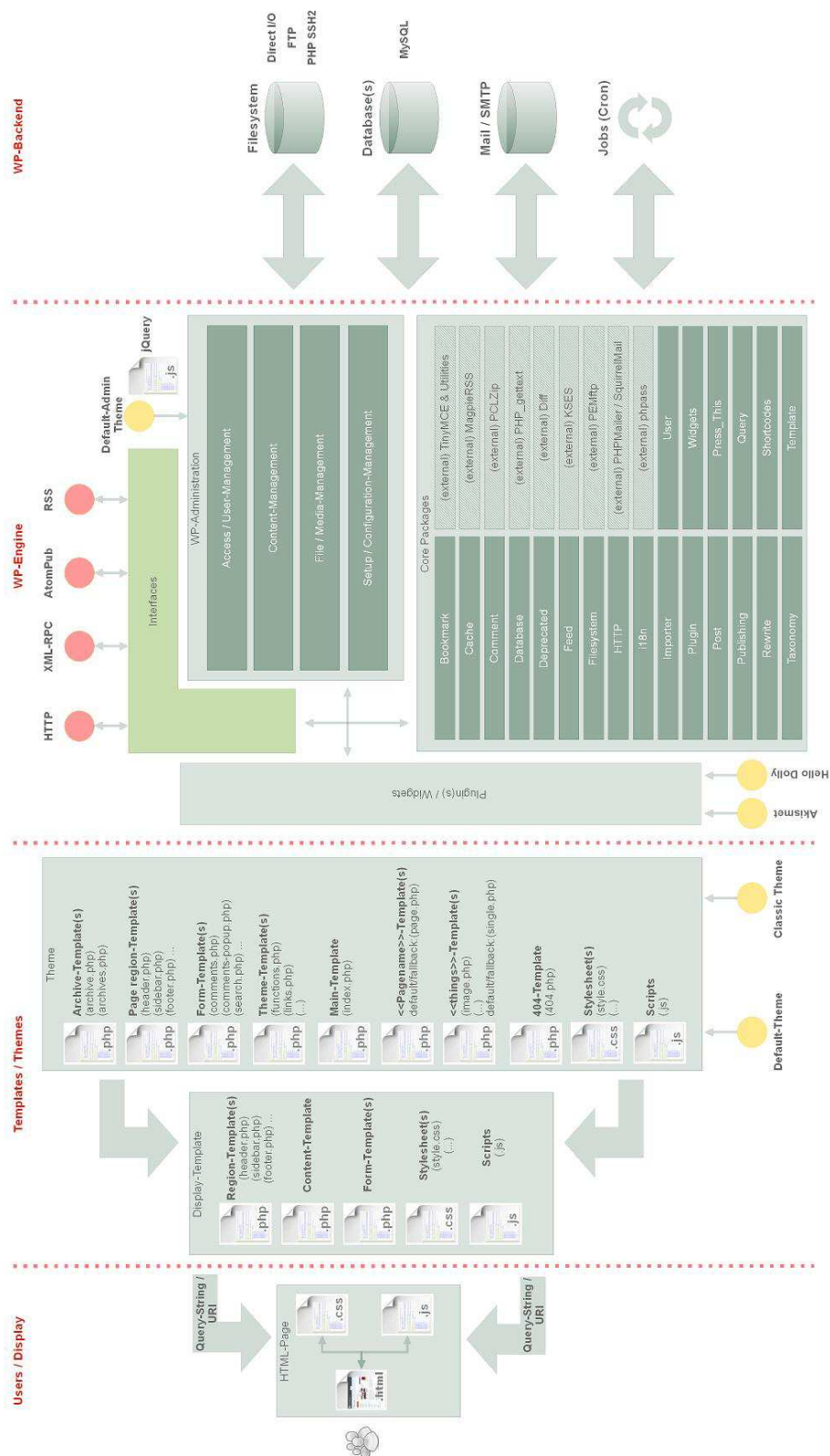
Příloha 1 na CD/DVD - Fixní layout

Příloha 2 na CD/DVD - Responzivní layout

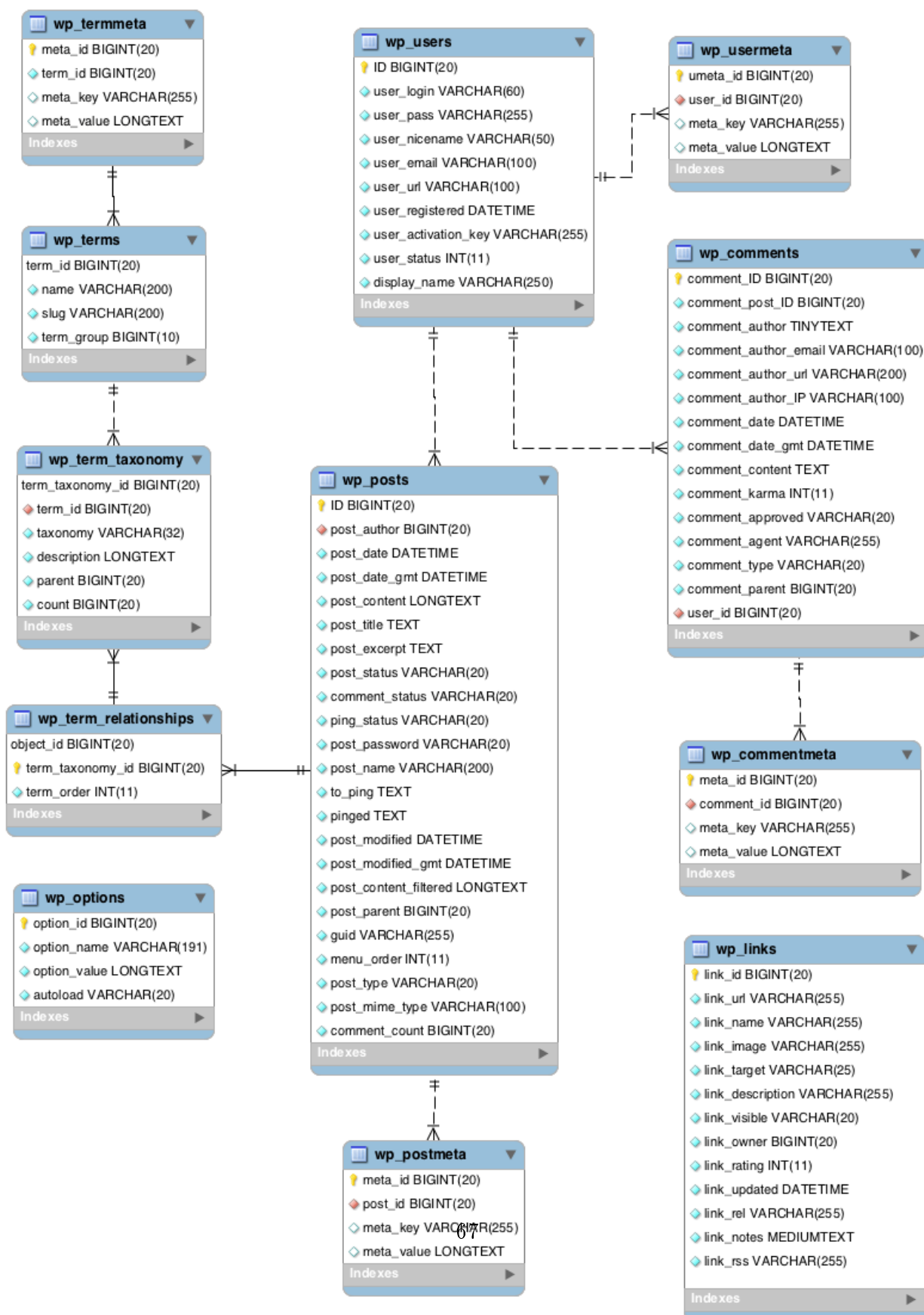
Příloha 3 na CD/DVD - WordPress šablona

Příloha 4 na CD/DVD - Ukázkové pluginy

# Příloha A - Podrobná architektura WordPressu



## Příloha B - Databázový model WordPressu



## Příloha C - Bootstrap komponenty

### Email address

### Password

### File input

 Soubor nevybrán

Example block-level help text here.

☐ Check me out

### *Formulář*

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

### *Tabulka*

Left

Center

Right

Justify

Default

Danger

Success

### *Skupina tlačítek a tlačítka*

Inbox 42

Messages 4

### *Indikátor počtu notifikací*





**Jsem odsazený text s obrázkem**

Pomocí media objectu je možné přidat k odsazenému textu obrázek, video či audio.



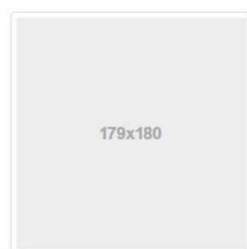
**Příspěvky**

Tato komponenta je vhodná především pro komentáře či příspěvky z Twitteru.

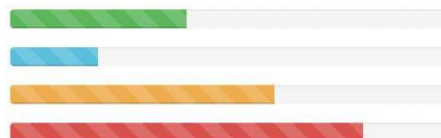
*Média objekty*

New	12
Deleted	5
Warnings	3

*Seznam skupin*



*Náhled obrázku*



*Indikátor průběhu*



*Štítky*

**Záhlaví stránky** Podnadpis pro záhlaví

*Záhlaví stránky*

Well done! You successfully read this important alert message.

Heads up! This alert needs your attention, but it's not super important.

Warning! Better check yourself, you're not looking too good.

Oh snap! Change a few things up and try submitting again.

### *Upozornění*

#### Nadpis panelu

Sem patří obsah panelu. Panel slouží k oddělení obsahu a členění textu do kapitol.

Patička panelu.

### *Panel*

## Ahoj, světe!

Jumbotron slouží pro marketingové nebo informační účely. Obsahuje upozorňující nadpis a informační obsah.

[Zjistit více](#)

### *Jumbotron*

140 x 140

140 x 140

140 x 140

### *Styly obrázků*

Nadpis vyskakovacího okna

Obsah vyskakovacího okna.

Zavřít

Uložit

*Vyskakovací okno (Modal)*



*Carousel*

Záložka 1

Záložka 2

Obsah v záložce 1

*Obsah oddělený v záložkách (Tab)*

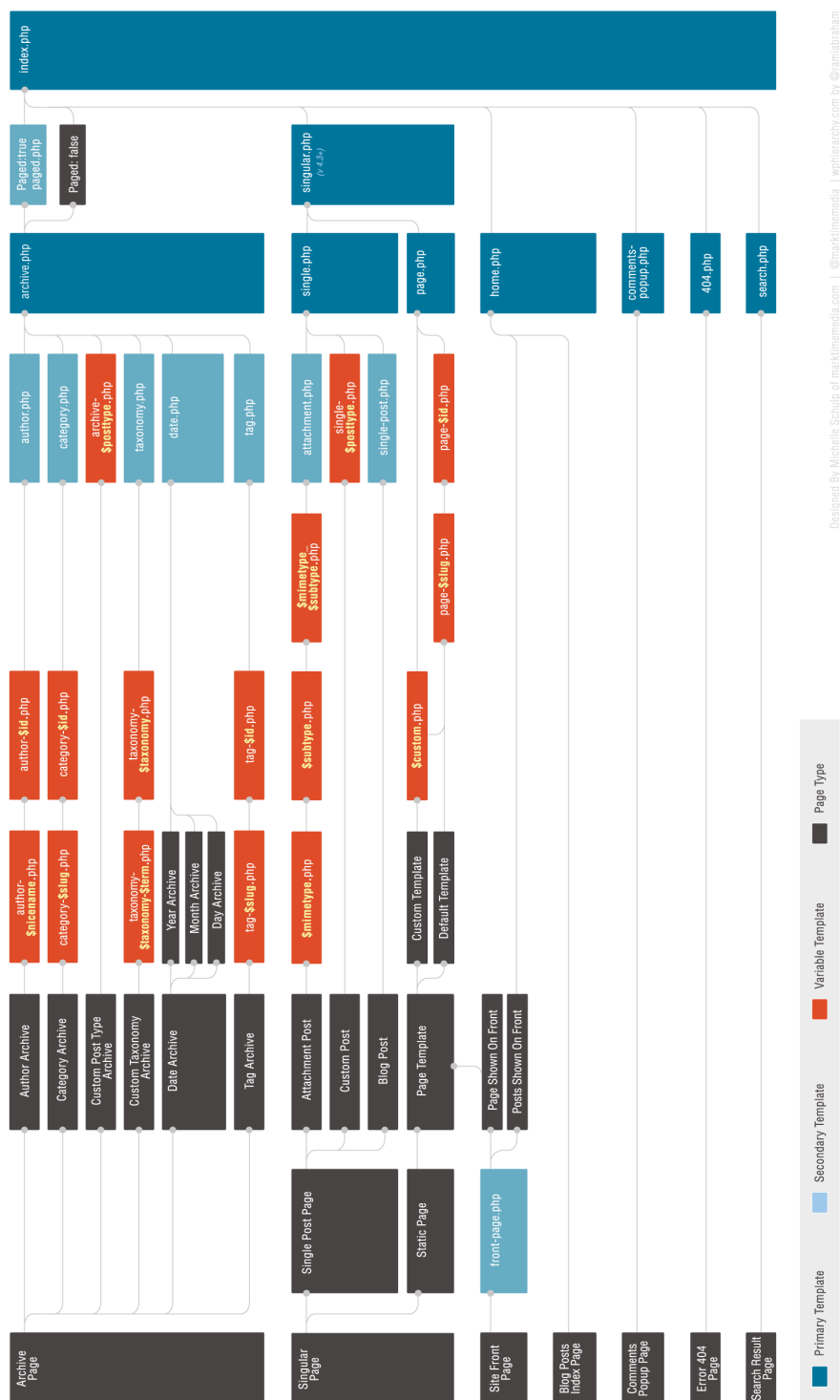
Home / Library / Data

*Drobečková navigace*


« 1 2 3 4 5 »


*Stránkování*

## Příloha D - Hierarchie WordPress šablon



## Příloha E - Výsledná podoba webového portálu



Hledany vyraz... 

Oficiální web Základní školy a Mateřské školy Těškovice

AKTUALITY

ZÁKLADNÍ ŠKOLA

MATEŘSKÁ ŠKOLA

ŠKOLNÍ DRUŽINA

ŠKOLNÍ JÍDELNA

O NÁS

KARIÉRA

HLAVNÍ MENU

Kariéra

O nás

Základní škola

Mateřská škola

Školní družina


Školní jídelna

Ukázka obarvení zdrojového kódu

27.04.2017

Ostatní

Václav Dobeš

 Upravit


```
0. function barevny_zdrojak_style(){
1.     global $barevny_zdrojak_codesnippetmatch_arrays;
2.     global $barevny_zdrojak_run_token;
3.     global $barevny_zdrojak_comments;
4.     global $barevny_zdrojak_used_languages;
5.     global $barevny_zdrojak_requested_css_files;
6.     $barevny_zdrojak_requested_css_files = array();
7.     $barevny_zdrojak_comments = array();
8.     $barevny_zdrojak_used_languages = array();
9.
10.    $barevny_zdrojak_run_token = uniqid(rand());
11.    barevny_zdrojak_filter_and_replace_code_snippets();
12.    if (!count($barevny_zdrojak_codesnippetmatch_arrays)) return;
13. }
```


POPIS OBLÍBENÉ HRAČKY

12.04.2017

akce ZŠ

admin

 Upravit



V hodinách slohu dostali žáci druhého a třetího ročníku za úkol popsat svou oblíbenou hračku. [\(Pokračování textu...\)](#)







## Oficiální web Základní školy a Mateřské školy Těškovice



### HLAVNÍ MENU

[Kariéra](#)[O nás](#)[Základní škola](#)[Mateřská škola](#)[Školní družina](#)[Školní jídelna](#)

### VELIKONOCE V ŽEROTÍNSKÉM ZÁMKU

 09.04.2017  akce ZŠ  admin  Upravit



Ve středu 5. dubna se žáci základní školy účastnili interaktivního programu na Žerotínském zámku v Novém Jičíně. „Neseme, neseme májiček“ byl zaměřený na tradice spjaté s příchodem jara a Velikonocemi. ([Pokračování textu...](#))